**JSDOM: Assignment**

You are building a contact book. The list of contacts and relevant details are stored in an array named contacts. You wish to create a few helper functions to display the contacts, filter by city and make a few modifications.

**Existing Files**

| **file path** | **description** |
| --- | --- |
| index.html | It contains placeholder elements for the list of contacts to be rendered. It also contains a form for filtering the list of contacts. *You should not edit the HTML file in any way*. |
| index.css | A few basic styles. This challenge does not focus on style. You may ignore this file. |
| index.js | You need to write your JavaScript code in this file. This is the only file that you need to edit. You will find the contacts array declared in this file. |
| test/solution.test.js | The tests your code will run against. You do not need to edit this file. |

**Tasks**

The contacts array contain contacts in the following format:

{

id: 39,

name: "Arwen Undómiel",

email: "arwen@imladris.com",

picture: "https://via.placeholder.com/150/FE0F0B/FFFFFF?Text=AU",

address: {

street: "Main Street",

suite: "Suite 17",

city: "Imladris",

zipcode: "90566-7771",

geo: {

lat: "-43.9509",

lng: "-34.4618",

},

},

phone: "010-692-6593 x09125",

website: "arwen.com",

company: {

name: "Eriador Services",

catchPhrase: "Proactive didactic contingency",

bs: "synergize scalable supply-chains",

},

},

Write the following functions in the index.js file.

**renderContact()**

This function accepts a contact object in the format described above.  
Create and return the HTML to render a single contact. The HTML for a single contact should look like this:

<div class="card" data-id="39">

<button class="deleteBtn" title="Delete this contact">X</button>

<div class="avatar">

<div class="circle"></div>

<div class="circle"></div>

<img src="https://via.placeholder.com/150/FE0F0B/FFFFFF?Text=AU" />

</div>

<div class="info">

<span class="name big">Arwen Undómiel</span>

<span class="email small">arwen@imladris.com</span>

</div>

<div class="details">

<div class="phone">010-692-6593 x09125</div>

<div class="website">arwen.com</div>

</div>

<div class="additional">

<div class="address">

<div class="suite">Suite 17</div>

<div class="street">Main Street</div>

<div class="city">Imladris, 90566-7771</div>

</div>

<div class="company">

<div class="label">Works at</div>

<div class="company-name">Eriador Services</div>

</div>

</div>

</div>

*The tests are looking for these specific elements and class names to be used. Be sure to check your spelling properly.*

**loadCities()**

Accepts an array of contacts.  
The load cities function finds all unique cities in the contact list and inserts a set of options in the select element with the id filterOptions. The default option should remain in place.

After populating the filterOptions select it should look like this:

<select name="filterOptions" id="filterOptions">

<option value="0">-- Select a city --</option>

<option value="New York">New York</option>

<option value="Kinsasha">Kinsasha</option>

</select>

**render()**

This method accepts an array of contacts.  
Render the array of contacts and insert them into the DOM.

Additionally, call the loadCities() function to ensure that the list of cities in the filter is updated.

The contacts should be rendered in the section with id "contacts". If there are no contacts in the list simply leave the section empty.

**filterByCity()**

Accepts a string representing the name of a city. Filter the list of contacts by contacts that live in the given city. Create a new array for the filtered contact. DO NOT MODIFY THE ORIGINAL CONTACTS ARRAY. Return the filtered contacts.

**filterHandler()**

This function attaches a change event listener to the filterOptions select. On change get the value of the select that was selected. If the value is "0" then call render() with the full list of contacts. Otherwise, call filterByCity() with the city that the user selected, then call render with the filtered list produced.

**deleteContact()**

Accepts an id representing the id of a contact in the contact list. Remove the contact from the contact list with the given id. If the id does not exist then no contacts are deleted.

**deleteButtonHandler()**

Add a click event handler to the deleteBtn elements.  
When clicked, get the id of the card that was clicked from the  
corresponding data-id then call deleteContact() and re-render  
the list.

**main()**

This is the starting point of the code. Call the necessary functions to add event listeners then render the contacts.