Lab Activity – Web Storage

Introduction

In this lab, you will implement the capability of saving the shopping list items entered by the user in our Shopping List program, by using local web storage. In version 1 of this program (attached to this document), this program was able to accept new shopping items and display them on the webpage. However, as soon as you close your browser or refresh the current page, those items disappeared. To avoid that, we will use local web storage to persist this data, so that when we revisit the webpage they are still listed in the page.

The major functionality for Version 2 of this program is:

- As the page finish loading its components, your JavaScript program shall retrieve the shopping items saved in the local storage and updates area that shows the list of items on the webpage. This functionality is not provided in version 1. You need to implement this functionality for Version 2.
- As new items are created by the user, these items must be displayed in the webpage (this functionality is already implemented) and also saved into the local storage (you need to implement this functionality).
- As old items are deleted by the user, these items must be deleted from the webpage (this functionality is already implemented) and these items must be removed from the local storage (you need to implement this functionality).
- By using local storage, it is expected that when the browser is closed and reopened, the page displays the items that were saved in the local storage.

Note: the code provided here for version 1 has lots of comments giving you some directions on what/how you should implement.

Original Code:

File: shopping_list_without_storage.html

```
<!DOCTYPE html>
<html>

<head>
    <title>Another DOM Example</title>
    <script type="text/javascript" src="shopping_list_without_storage.js"> </script>
</head>

<body>
    <h1>Shopping List</h1>
    <div>
        <input type="button" id="addNewItemButton" value="Add New Item">
        <input type="button" id="selectAllItemsButton" value="Select All Items">
        <input type="button" id="deleteSelectedItemsButton" value="Delete Selected
Items">
    </div>
    <br>

    <div id="inputForNewItem"></div>
    <br>
    <div id="listOfItems"></div>

</body>
</html>
```

File: shopping_list_without_storage.js

```
// Run setup after page had loaded
window.addEventListener("load", setup);

function setup() {
    addListeners();
    // const data = retrieveItemsFromLocalStorage();  ← uncomment this for Version 2
    // populateShoppingList(data);   ← uncomment this for Version 2
}

// add listeners to the buttons
function addListeners() {
    document.getElementById("addNewItemButton").addEventListener("click", addNewItem);
    document.getElementById("deleteSelectedItemsButton").addEventListener("click",
deleteSelectedItems);
    document.getElementById("selectAllItemsButton").addEventListener("click",
selectAllItems);
}

// function retrieveItemsFromLocalStorage() {      ← uncomment this function for Version 2
//     create an empty data array. This array will be populated
//     with item objects from the local storage
//     An item object will have only one pair of (key,value)
//     The "key" will be the item description, such as "banana",
//     and the "value" will be the quantity, such as "2 dozen"

//     loop over all the local storage items using a FOR loop
//     retrive the key,value, create an object of them and
//     add this object (push command) into the data array discribed
//     above

//     // return the array with shopping item objects
// }

// my gift to you. I am giving this function ready to use   ← uncomment this for Version 2
// function populateShoppingList(data) {
//     for (datum of data) {
//         // there is only one pair of key, value per item object
//         // this for loop is just to retrieve them
//         for (var [key, value] of Object.entries(datum)) { }
//         addItemToShoppingListArea(key, value);
//     }
// }

function addItemToShoppingListArea(key, value) {
    // at this point there is only one key/value in datum object
    const divElement = document.createElement("div");

    // creating the checkbox
    const listItemCheckBoxElement = document.createElement("input");
    listItemCheckBoxElement.setAttribute("type", "checkbox");
    listItemCheckBoxElement.setAttribute("name", "checkBoxName");
    listItemCheckBoxElement.setAttribute("class", "checkBoxClass");

    // create the label that goes with the checkbox
    const checkBoxLabelElement = document.createElement("label");
    checkBoxLabelElement.setAttribute("for", "checkBoxName");
```

```
    // getting the text that will be placed on the label
    checkBoxLabelElement.textContent = `${key} - ${value}`;
    listItemCheckBoxElement.setAttribute("id", key);

    // place a <br>. It could be done via CSS
    const breakElement = document.createElement("br");

    // appending the new elements into the <div>
    divElement.appendChild(listItemCheckBoxElement);
    divElement.appendChild(checkBoxLabelElement);
    divElement.appendChild(breakElement);

    // appending the new <div> into the existing <div> with id listOfItems
    const listDivElement = document.getElementById("listOfItems");
    listDivElement.appendChild(divElement);
}

// create an input text field and an associate button to enter new item
// into the shopping list
function addNewItem() {
    // creating the div element
    const divElement = document.createElement("div");

    // creating the input text element
    const itemInputTextElement = document.createElement("input");
    itemInputTextElement.setAttribute("type", "text");
    itemInputTextElement.setAttribute("size", 15);
    itemInputTextElement.setAttribute("id", "newItemDescription");

    const quantityInputTextElement = document.createElement("input");
    quantityInputTextElement.setAttribute("type", "text");
    quantityInputTextElement.setAttribute("size", 15);
    quantityInputTextElement.setAttribute("id", "newItemQuantity");

    // creating a button to add the new shopping item into the list the text entered
    const addNewButton = document.createElement("input");
    addNewButton.setAttribute("type", "button");
    addNewButton.setAttribute("id", "addNewButtonID");
    addNewButton.setAttribute("value", "Add New Item");
    addNewButton.addEventListener("click", addNewItemToTheShoppingList);

    // appending elements into the DOM under the already existing <div> (see HTML file)
    const itemListDiv = document.getElementById("inputForNewItem");
    itemListDiv.appendChild(itemInputTextElement);
    itemListDiv.appendChild(quantityInputTextElement);
    itemListDiv.appendChild(addNewButton);
}
```

```
function addNewItemToTheShoppingList() {
    // getting the text that will be placed on the label
    const itemTextDescriptionElement = document.getElementById("newItemDescription");
    const quantityInputTextElement = document.getElementById("newItemQuantity");
    addItemToShoppingListArea(itemTextDescriptionElement.value,
quantityInputTextElement.value);

    // removing the input field and the button associated with entering the new item
    const itemListDiv = document.getElementById("inputForNewItem");

    // removing the temporary fields to enter the new data
    itemListDiv.removeChild(itemTextDescriptionElement);
    itemListDiv.removeChild(quantityInputTextElement);
    itemListDiv.removeChild(document.getElementById("addNewButtonID"));

    // adding item in the local storage
    //
}

function selectAllItems() {
    // getting a list of the DOM elements that have class name checkBoxClass
    const checkBoxList = document.getElementsByClassName("checkBoxClass");

    // loop over the list and set the checkbox to true
    for (let i = 0; i < checkBoxList.length; i++) {
        checkBoxList[i].checked = true;
    }
}

function deleteSelectedItems() {
    // getting a list of the DOM elements that have class name checkBoxClass
    const checkBoxList = document.getElementsByClassName("checkBoxClass");

    // looping over the list and removing the parent <div>.
    // by doing so, the checkbox and the associated label are removed
    for (let i = checkBoxList.length - 1; i >= 0; i--) {
        if (checkBoxList[i].checked === true) {

            // add command to remove the item from the local storage here

            // removing the deleted item from the shopping list area
            checkBoxList[i].parentElement.remove();
        }
    }
}
```