

- the main idea of the project is to create page that contains form that enable us to create a list of items related to the grocery , which contains some functions such as add items to the list, delete single item from the list, delete all items from the list , and update storage after all operations.

```
<!DOCTYPE html>
<html>
<head>
  <link href="css/bootstrap.min.css" rel="stylesheet" >
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
  <link rel="preconnect" href="https://fonts.gstatic.com">
<script src="tata.js"></script>
  <link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;600;700&display=swap" rel="stylesheet">
<style>
  *{
    font-family: cairo;
    font-weight: bold;
  }
</style>
<title>Grocery List bay omar nasr,mokhtar,ahmed,ibrahim</title>
```

### **First in the head section :**

1. make the page is responsive to the bootstrap through bring up bootstrap files
2. bring up fonts file and make the page fonts bold and cairo.

3. declare script "tata" which contains functions that show alarm message to the user

### **Second in the body section :**

```
4. class="bg-secondary"
```

contains all page and make its color gray ,

```
class="container-fluid p-3">
```

container-fluid class provides a full width container, spanning the entire width of the viewport,

p-3 means that make page responsive to bootstrap when open it on different screens ,

```
class="row justify-content-center">
```

make the content of page at the center

```
class="card bg-info text-center col-lg-6 col-md-12"
```

cards provide a flexible and extensible content container with multiple variants and options,

```
<h5 class="card-header">Add New Item</h5>
```

Make a text area to take a value,

```
<form action="">
  <input autocomplete="off" type="text" id="description" class="
form-control text-center">
  <a onclick="addToDo()" class="btn btn-success btn-lg mt-
5">Add
    <i class="fa fa-plus-circle"></i>
  </a>
```

form to take text value and submit it to list when click on the button add+ the value which saved by id=description will added to the next container in form of a table ,

```
<div class="row justify-content-center">

  <div class="card bg-dark text-center col-6 mt-4">
    <h5 class="card-header text-white">Items</h5>
    <div class="card-body">
      <a onclick="removeItems()" class="btn btn-danger btn-lg mt-1">Clear
        <i class="fa fa-plus-trash"></i>
      </a>
      <table class="table table-dark table-striped table-bordered">
        <thead>
        </thead>
        <tbody class="tbody-items">
```

when new value came to the next container saved in table and make a trash button which enable us to delete a single item from the table and make a clear button which can delete all items from the list .

```
<script src="app.js"></script>
```

At the last bring up script app.js which contains all function we have declared above to make the page responsive to all operations .

### **Third the functions used in app.js :**

```
const txt_description = document.getElementById("description")
const list = document.querySelector('.tbody-items');
displayStorage()
function addToDo() {
  var value = txt_description.value;
  if (value === '') {
```

```

        tata.error('Error', 'Please add grocery item')
    } else {
        createItem(value);
        tata.info(`${value}`, 'Has Been Added To List')
        updateStorage(value);
    }

```

This function take value of id and put it in table when click on the button add+ , and when there is no text to added show a message “please add grocery item “ and when there is new item added to the list show message “this item “ milk” has been add to list”

```

function removeItems() {
    //delete from local storage
    localStorage.removeItem('items');
    let items = document.querySelectorAll('.groceryitem');
    ///tbody-items
    if (items.length == 0) {
        tata.error('Error', "there's no items to delete")
    }
    else{
        document.querySelector('tbody-items').innerHTML = "";
        tata.error('Deleted', "All Items Have Been Deleted")
    }
}

```

This function select all items from the list then delete them , first make the local storage empty then if there is items in the list select all and delete them and show message “all items have been deleted “

And if there is no items in the list show message  
“there is no items to delete “

```
}  
// create item  
function createItem(value) {  
    let parent = document.createElement('tr');  
    parent.innerHTML = ` ${value}</td><td> <a href = "##" onclick="removeSingleItem()" class="btn-delete"> <span class="fa fa- trash text-danger"></span> </a></td>`;       list.appendChild(parent); } |
```

This function create an item in the list by creating a row in the table and put in it the value came from first function which called add+ and make a button which called btn to link it with another function to enable us delete this single value in the row

```
}  
function displayStorage() {  
    let exists = localStorage.getItem('items');  
    if (exists) {  
        let storageItems = JSON.parse(localStorage.getItem('items'));  
        storageItems.forEach(function (element) {  
            createItem(element);  
        })  
    }  
}
```

This function take every item in the list after added to list to store and show it in the local storage in the form of json.parse and this is a way to save items in order in the storage

```
function removeSingleItem() {  
    let link = this.event.target.parentElement;
```

```

    link.parentElement.parentElement.remove()
    groceryList = localStorage.getItem('items') ? JSON.parse(localStorage.getItem('items')) : [];
    var currentindex = link.parentElement.parentElement.rowIndex;

    groceryList.splice(currentindex, 1);
    localStorage.setItem('items', JSON.stringify(groceryList));
    tata.warn(`${link.parentElement.parentElement.children[0].innerText}`, 'Deleted')
}

```

This function select the parent of the value we want to delete which mean the row which contain this value then delete it , and delete the index of this value from the local storage also and show message “this item deleted “

```

function updateStorage(value) {
    let groceryList;
    groceryList = localStorage.getItem('items') ? JSON.parse(localStorage.getItem('items')) : [];
    groceryList.push(value);
    localStorage.setItem('items', JSON.stringify(groceryList));
}

```

After every operation we should update the local storage such if we add 3 items in the list and we want to add new item , this will be added after the last item then update storage , this is the form of Json and stringify functions save the elements in order .

Team leader :omar nasr elsayed