**Web Technology**

**Naresh Kumar**

**22IT3025**

**Lab-7**

T1 :

Solution ::

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Shopping List (Same after reload)</title>

<style>

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

margin: 20px;

background-color: #f4f4f4;

color: #333;

}

#shopping-list {

max-width: 800px;

margin: 0 auto;

background-color: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 {

text-align: center;

color: #4285f4;

}

#item-type, #item-dropdown {

margin-bottom: 10px;

width: 100%;

padding: 8px;

box-sizing: border-box;

border: 1px solid #ccc;

border-radius: 4px;

}

#quantity {

margin-bottom: 10px;

width: 70%;

padding: 8px;

box-sizing: border-box;

border: 1px solid #ccc;

border-radius: 4px;

}

button {

padding: 8px 12px;

background-color: #4285f4;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #357ae8;

}

.table-container {

display: flex;

justify-content: space-between;

align-items: flex-start;

margin-top: 20px;

}

.table-container div {

width: 48%;

}

h2 {

color: #4285f4;

}

ul {

list-style-type: none;

padding: 0;

margin: 0;

}

li {

margin-bottom: 5px;

display: flex;

align-items: center;

}

label {

margin-right: 10px;

}

table {

width: 100%;

margin-top: 10px;

border-collapse: collapse;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

.purchased-item {

text-decoration: line-through;

}

#grand-total {

margin-top: 10px;

font-weight: bold;

color: #4285f4;

}

</style>

</head>

<body>

<div id="shopping-list">

<h1>Shopping List</h1>

<div>

<label for="item-type">Select Item Type:</label>

<select id="item-type">

<option value="grocery">Grocery</option>

<option value="clothing">Clothing</option>

<option value="electronics">Electronics</option>

</select>

</div>

<div>

<label for="item-dropdown">Select Item:</label>

<select id="item-dropdown"></select>

<input type="number" id="quantity" placeholder="Quantity" min="1" value="1">

<button onclick="addItem()">Add Item</button>

</div>

<div class="table-container">

<div>

<h2>Items To Buy</h2>

<table id="items-to-buy-table">

<thead>

<tr>

<th>Item</th>

<th>Item Type</th>

<th>Quantity</th>

<th>Price (Rs.)</th>

<th>Mark as Purchased</th>

</tr>

</thead>

<tbody id="items-list"></tbody>

</table>

<div id="grand-total"></div>

</div>

<div>

<h2>Purchased Items</h2>

<table id="purchased-items-table">

<thead>

<tr>

<th>Item</th>

<th>Item Type</th>

<th>Quantity</th>

<th>Price (Rs.)</th>

<th>Remove</th>

</tr>

</thead>

<tbody id="purchased-items-list"></tbody>

</table>

</div>

</div>

</div>

<script>

const predefinedItems = {

grocery: [

{ name: "Bread", price: 150.00 },

{ name: "Milk", price: 80.00 },

{ name: "Eggs", price: 60.00 }

],

clothing: [

{ name: "T-shirt", price: 750.00 },

{ name: "Jeans", price: 1200.00 },

{ name: "Socks", price: 99.00 }

],

electronics: [

{ name: "Laptop", price: 35000.00 },

{ name: "Headphones", price: 1299.00 },

{ name: "Charger", price: 899.00 }

]

};

const itemTypeSelect = document.getElementById('item-type');

const itemDropdown = document.getElementById('item-dropdown');

const itemsList = document.getElementById('items-list');

const purchasedItemsList = document.getElementById('purchased-items-list');

loadShoppingList();

function loadShoppingList() {

const savedItemsToBuy = localStorage.getItem('itemsToBuy');

const savedPurchasedItems = localStorage.getItem('purchasedItems');

if (savedItemsToBuy) {

itemsList.innerHTML = savedItemsToBuy;

}

if (savedPurchasedItems) {

purchasedItemsList.innerHTML = savedPurchasedItems;

}

updateGrandTotal();

}

function saveShoppingList() {

const itemsToBuy = itemsList.innerHTML;

const purchasedItems = purchasedItemsList.innerHTML;

localStorage.setItem('itemsToBuy', itemsToBuy);

localStorage.setItem('purchasedItems', purchasedItems);

}

itemTypeSelect.addEventListener('change', updateItemDropdown);

updateItemDropdown();

function updateItemDropdown() {

const selectedItemType = itemTypeSelect.value;

const items = predefinedItems[selectedItemType];

itemDropdown.innerHTML = '';

items.forEach(item => {

const option = document.createElement('option');

option.value = item.name;

option.textContent = item.name;

itemDropdown.appendChild(option);

});

}

function addItem() {

const selectedItemType = itemTypeSelect.value;

const selectedItemName = itemDropdown.value;

const quantity = parseInt(document.getElementById('quantity').value, 10);

if (!isNaN(quantity) && quantity > 0) {

const itemPrice = getItemPrice(selectedItemType, selectedItemName);

const totalPrice = quantity \* itemPrice;

const row = document.createElement('tr');

row.innerHTML = `

<td>${selectedItemName}</td>

<td>${selectedItemType}</td>

<td>${quantity}</td>

<td>${itemPrice.toFixed(2)}</td>

<td><input type="checkbox" onchange="markAsPurchased(this, '${selectedItemName}', '${selectedItemType}', ${itemPrice}, ${quantity})"></td>

`;

itemsList.appendChild(row);

document.getElementById('quantity').value = '';

updateGrandTotal();

saveShoppingList();

}

}

function markAsPurchased(checkbox, itemName, itemType, itemPrice, quantity) {

const row = checkbox.parentElement.parentElement;

if (checkbox.checked) {

row.classList.add('purchased-item');

addToPurchasedItems(itemName, itemType, itemPrice, quantity);

row.remove();

} else {

row.classList.remove('purchased-item');

}

updateGrandTotal();

saveShoppingList();

}

function addToPurchasedItems(itemName, itemType, itemPrice, quantity) {

const row = document.createElement('tr');

row.innerHTML = `

<td>${itemName}</td>

<td>${itemType}</td>

<td>${quantity}</td>

<td>${(itemPrice \* quantity).toFixed(2)}</td>

<td><button onclick="removePurchasedItem(this.parentElement.parentElement)">Remove</button></td>

`;

purchasedItemsList.appendChild(row);

updateGrandTotal();

saveShoppingList();

}

function removePurchasedItem(row) {

row.remove();

updateGrandTotal();

saveShoppingList();

}

function updateGrandTotal() {

const grandTotalDiv = document.getElementById('grand-total');

const items = document.querySelectorAll('#items-list tr:not(.purchased-item)');

let grandTotal = 0;

items.forEach(item => {

grandTotal += parseFloat(item.cells[3].textContent);

});

grandTotalDiv.textContent = `Grand Total: Rs. ${grandTotal.toFixed(2)}`;

}

function getItemPrice(itemType, itemName) {

const items = predefinedItems[itemType];

const selectedItem = items.find(item => item.name === itemName);

return selectedItem ? selectedItem.price : 0;

}

</script>

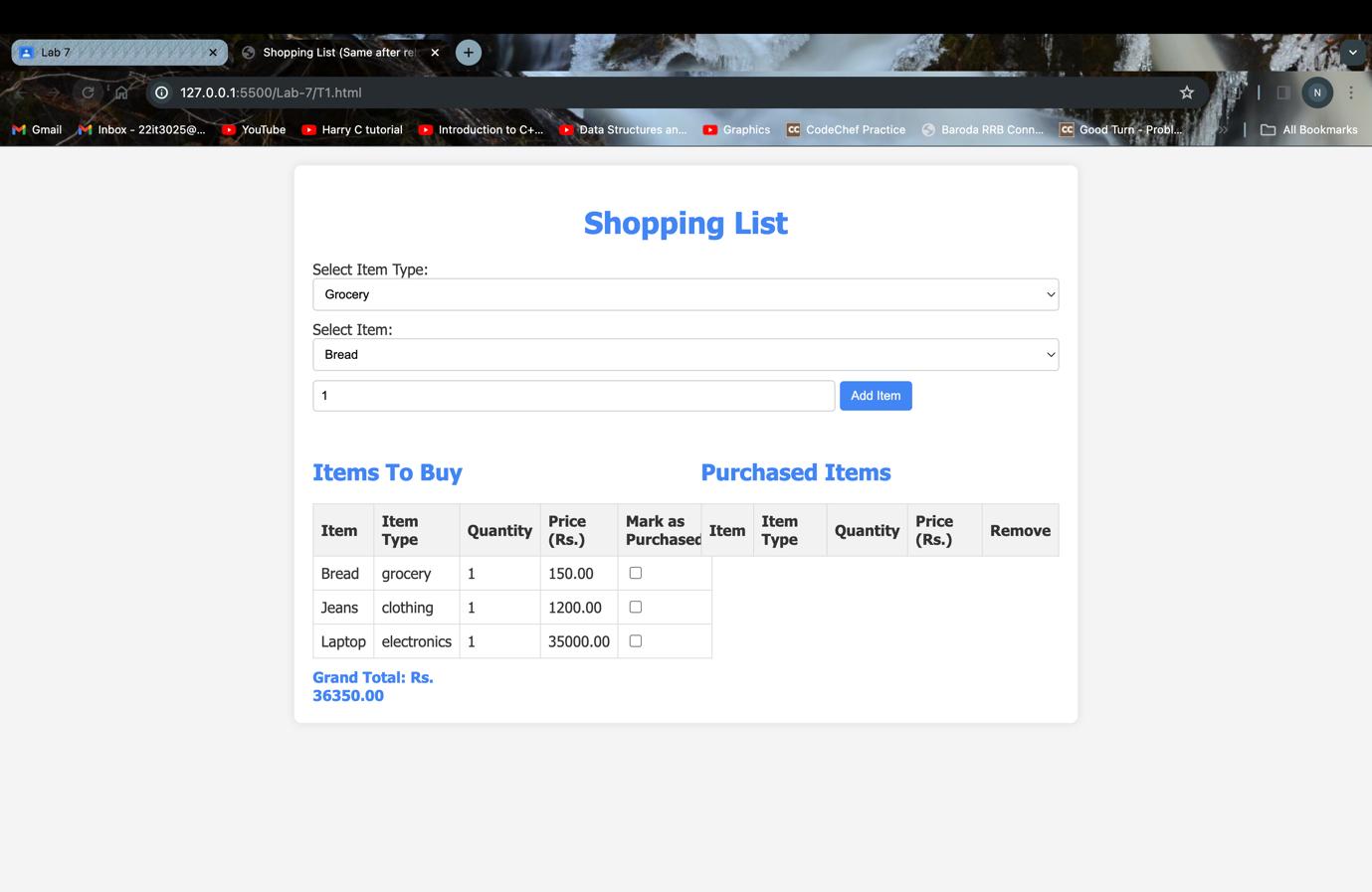
</body>

</html>

OUTPUT

A screenshot of a computer

Description automatically generated

After the reload 

T2 :

Solution ::

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Shopping List (Same after reload)</title>

<style>

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

margin: 20px;

background-color: #f4f4f4;

color: #333;

}

#shopping-list {

max-width: 800px;

margin: 0 auto;

background-color: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 {

text-align: center;

color: #4285f4;

}

#item-type, #item-dropdown {

margin-bottom: 10px;

width: 100%;

padding: 8px;

box-sizing: border-box;

border: 1px solid #ccc;

border-radius: 4px;

}

#quantity {

margin-bottom: 10px;

width: 70%;

padding: 8px;

box-sizing: border-box;

border: 1px solid #ccc;

border-radius: 4px;

}

button {

padding: 8px 12px;

background-color: #4285f4;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #357ae8;

}

.table-container {

display: flex;

justify-content: space-between;

align-items: flex-start;

margin-top: 20px;

}

.table-container div {

width: 48%;

}

h2 {

color: #4285f4;

}

ul {

list-style-type: none;

padding: 0;

margin: 0;

}

li {

margin-bottom: 5px;

display: flex;

align-items: center;

}

label {

margin-right: 10px;

}

table {

width: 100%;

margin-top: 10px;

border-collapse: collapse;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

.purchased-item {

text-decoration: line-through;

}

#grand-total {

margin-top: 10px;

font-weight: bold;

color: #4285f4;

}

</style>

</head>

<body>

<div id="shopping-list">

<h1>Shopping List</h1>

<div>

<label for="item-type">Select Item Type:</label>

<select id="item-type">

<option value="grocery">Grocery</option>

<option value="clothing">Clothing</option>

<option value="electronics">Electronics</option>

</select>

</div>

<div>

<label for="item-dropdown">Select Item:</label>

<select id="item-dropdown"></select>

<input type="number" id="quantity" placeholder="Quantity" min="1" value="1">

<button onclick="addItem()">Add Item</button>

</div>

<div class="table-container">

<div>

<h2>Items To Buy</h2>

<table id="items-to-buy-table">

<thead>

<tr>

<th>Item</th>

<th>Item Type</th>

<th>Quantity</th>

<th>Price (Rs.)</th>

<th>Mark as Purchased</th>

</tr>

</thead>

<tbody id="items-list"></tbody>

</table>

<div id="grand-total"></div>

</div>

<div>

<h2>Purchased Items</h2>

<table id="purchased-items-table">

<thead>

<tr>

<th>Item</th>

<th>Item Type</th>

<th>Quantity</th>

<th>Price (Rs.)</th>

<th>Remove</th>

</tr>

</thead>

<tbody id="purchased-items-list"></tbody>

</table>

</div>

</div>

</div>

<script>

const predefinedItems = {

grocery: [

{ name: "Bread", price: 150.00 },

{ name: "Milk", price: 80.00 },

{ name: "Eggs", price: 60.00 }

],

clothing: [

{ name: "T-shirt", price: 750.00 },

{ name: "Jeans", price: 1200.00 },

{ name: "Socks", price: 99.00 }

],

electronics: [

{ name: "Laptop", price: 35000.00 },

{ name: "Headphones", price: 1299.00 },

{ name: "Charger", price: 899.00 }

]

};

const itemTypeSelect = document.getElementById('item-type');

const itemDropdown = document.getElementById('item-dropdown');

const itemsList = document.getElementById('items-list');

const purchasedItemsList = document.getElementById('purchased-items-list');

loadShoppingList();

function loadShoppingList() {

const savedItemsToBuy = localStorage.getItem('itemsToBuy');

const savedPurchasedItems = localStorage.getItem('purchasedItems');

if (savedItemsToBuy) {

itemsList.innerHTML = savedItemsToBuy;

}

if (savedPurchasedItems) {

purchasedItemsList.innerHTML = savedPurchasedItems;

}

updateGrandTotal();

}

function saveShoppingList() {

const itemsToBuy = itemsList.innerHTML;

const purchasedItems = purchasedItemsList.innerHTML;

localStorage.setItem('itemsToBuy', itemsToBuy);

localStorage.setItem('purchasedItems', purchasedItems);

}

itemTypeSelect.addEventListener('change', updateItemDropdown);

updateItemDropdown();

function updateItemDropdown() {

const selectedItemType = itemTypeSelect.value;

const items = predefinedItems[selectedItemType];

itemDropdown.innerHTML = '';

items.forEach(item => {

const option = document.createElement('option');

option.value = item.name;

option.textContent = item.name;

itemDropdown.appendChild(option);

});

}

function addItem() {

const selectedItemType = itemTypeSelect.value;

const selectedItemName = itemDropdown.value;

const quantity = parseInt(document.getElementById('quantity').value, 10);

if (!isNaN(quantity) && quantity > 0) {

const itemPrice = getItemPrice(selectedItemType, selectedItemName);

const totalPrice = quantity \* itemPrice;

const row = document.createElement('tr');

row.innerHTML = `

<td>${selectedItemName}</td>

<td>${selectedItemType}</td>

<td>${quantity}</td>

<td>${itemPrice.toFixed(2)}</td>

<td><input type="checkbox" onchange="markAsPurchased(this, '${selectedItemName}', '${selectedItemType}', ${itemPrice}, ${quantity})"></td>

`;

itemsList.appendChild(row);

document.getElementById('quantity').value = '';

updateGrandTotal();

saveShoppingList();

}

}

function markAsPurchased(checkbox, itemName, itemType, itemPrice, quantity) {

const row = checkbox.parentElement.parentElement;

if (checkbox.checked) {

row.classList.add('purchased-item');

addToPurchasedItems(itemName, itemType, itemPrice, quantity);

row.remove();

} else {

row.classList.remove('purchased-item');

}

updateGrandTotal();

saveShoppingList();

}

function addToPurchasedItems(itemName, itemType, itemPrice, quantity) {

const row = document.createElement('tr');

row.innerHTML = `

<td>${itemName}</td>

<td>${itemType}</td>

<td>${quantity}</td>

<td>${(itemPrice \* quantity).toFixed(2)}</td>

<td><button onclick="removePurchasedItem(this.parentElement.parentElement)">Remove</button></td>

`;

purchasedItemsList.appendChild(row);

updateGrandTotal();

saveShoppingList();

}

function removePurchasedItem(row) {

row.remove();

updateGrandTotal();

saveShoppingList();

}

function updateGrandTotal() {

const grandTotalDiv = document.getElementById('grand-total');

const items = document.querySelectorAll('#items-list tr:not(.purchased-item)');

let grandTotal = 0;

items.forEach(item => {

grandTotal += parseFloat(item.cells[3].textContent);

});

grandTotalDiv.textContent = `Grand Total: Rs. ${grandTotal.toFixed(2)}`;

}

function getItemPrice(itemType, itemName) {

const items = predefinedItems[itemType];

const selectedItem = items.find(item => item.name === itemName);

return selectedItem ? selectedItem.price : 0;

}

function updateAccessCount() {

if (typeof(Storage) !== "undefined") {

let accessCount = localStorage.getItem('accessCount');

accessCount = accessCount ? parseInt(accessCount) + 1 : 1;

localStorage.setItem('accessCount', accessCount);

document.body.innerHTML += `<h1>You have visited this page ${accessCount} times.</h1>`;

} else {

document.body.innerHTML += "<h1>Sorry, your browser does not support localStorage.</h1>";

}

}

window.onload = updateAccessCount;

</script>

</body>

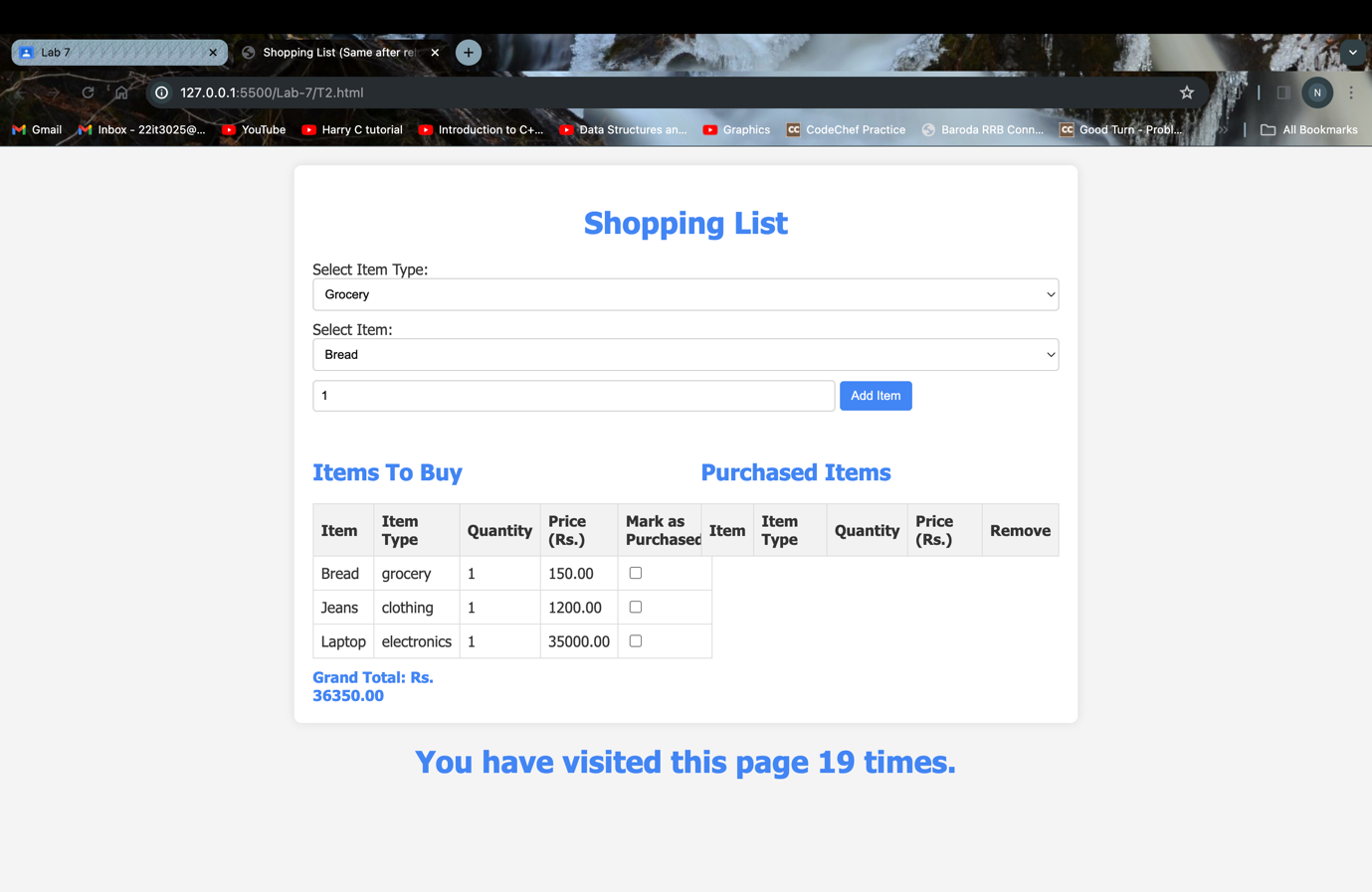
</html>

OUTPUT

A screenshot of a computer

Description automatically generated

After two reloads –



After one more reload

