T 1.

<!DOCTYPE html>

<html lang="en">

<head>

<title>Shopping List Application</title>

<link rel="stylesheet" href="css/shoppingstyle.css">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f8f8f8;

margin: 0;

padding: 0;

}

.container {

margin: 20px;

}

header {

background-color: #333;

color: white;

padding: 10px;

}

header h1 {

margin: 0;

}

nav ul {

list-style-type: none;

margin: 0;

padding: 0;

}

nav ul li {

display: inline;

margin-right: 10px;

}

.panel {

background-color: #fff;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

margin-bottom: 20px;

}

.panel-body {

padding: 20px;

}

.form-row {

margin-bottom: 20px;

}

.form-group {

margin-right: 10px;

}

.form-control {

width: 150px;

}

.btn {

background-color: #337ab7;

color: #fff;

padding: 8px 12px;

border: none;

cursor: pointer;

}

.btn:hover {

background-color: #286090;

}

.table-container {

max-width: 800px; /\* Adjust the max-width as needed \*/

margin: 0 auto;

}

table {

width: 100%;

border-collapse: collapse;

}

th, td {

padding: 10px;

border: 1px solid #ddd;

text-align: left;

}

th {

background-color: #f2f2f2;

}

</style>

</head>

<body>

<div class="container">

<header>

<h1>Shopping List</h1>

<nav>

<ul>

<li class="active"><a href="#">Home</a></li>

<li><a href="#">Page 1</a></li>

<li><a href="#">Page 2</a></li>

<li><a href="#">Page 3</a></li>

</ul>

</nav>

</header>

<div class="panel">

<div class="panel-body">

<div class="form-row">

<div class="form-group">

<label for="itemname">Item Name</label>

<input type="text" class="form-control" name="itemname" id="itemname">

</div>

<div class="form-group">

<label for="qty">Quantity</label>

<select name="quantity" id="qty" class="form-control">

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

<option value="7">7</option>

<option value="8">8</option>

<option value="9">9</option>

<option value="10">10</option>

</select>

</div>

<div class="form-group">

<label for="priority">Priority</label>

<select name="priority" id="priority" class="form-control">

<option value="low">low</option>

<option value="medium">medium</option>

<option value="high">high</option>

</select>

</div>

</div>

<div class="form-row">

<div class="form-group">

<label for="store">Store Name</label>

<select name="store" id="store" class="form-control">

<option value="Spencers">Spencers</option>

<option value="FUN">FUN</option>

<option value="Whole foods">Whole foods</option>

</select>

</div>

<div class="form-group">

<label for="category">Store Section</label>

<select name="category" id="category" class="form-control">

<option value="Meat">Meat</option>

<option value="Fruits">Fruits</option>

<option value="Vegetables">Vegetables</option>

</select>

</div>

<div class="form-group">

<label for="price">Price</label>

<input type="text" name="price" id="price" class="form-control">

</div>

</div>

<div class="form-row" style="float: right;">

<button onclick="clickedon()" class="btn">Add Item</button>

</div>

</div>

</div>

<div class="table">

<table>

<thead>

<tr>

<th></th>

<th>Item</th>

<th>Quantity</th>

<th>Store</th>

<th>Section</th>

<th>Price</th>

</tr>

</thead>

<tbody id="shoppinglist"></tbody>

</table>

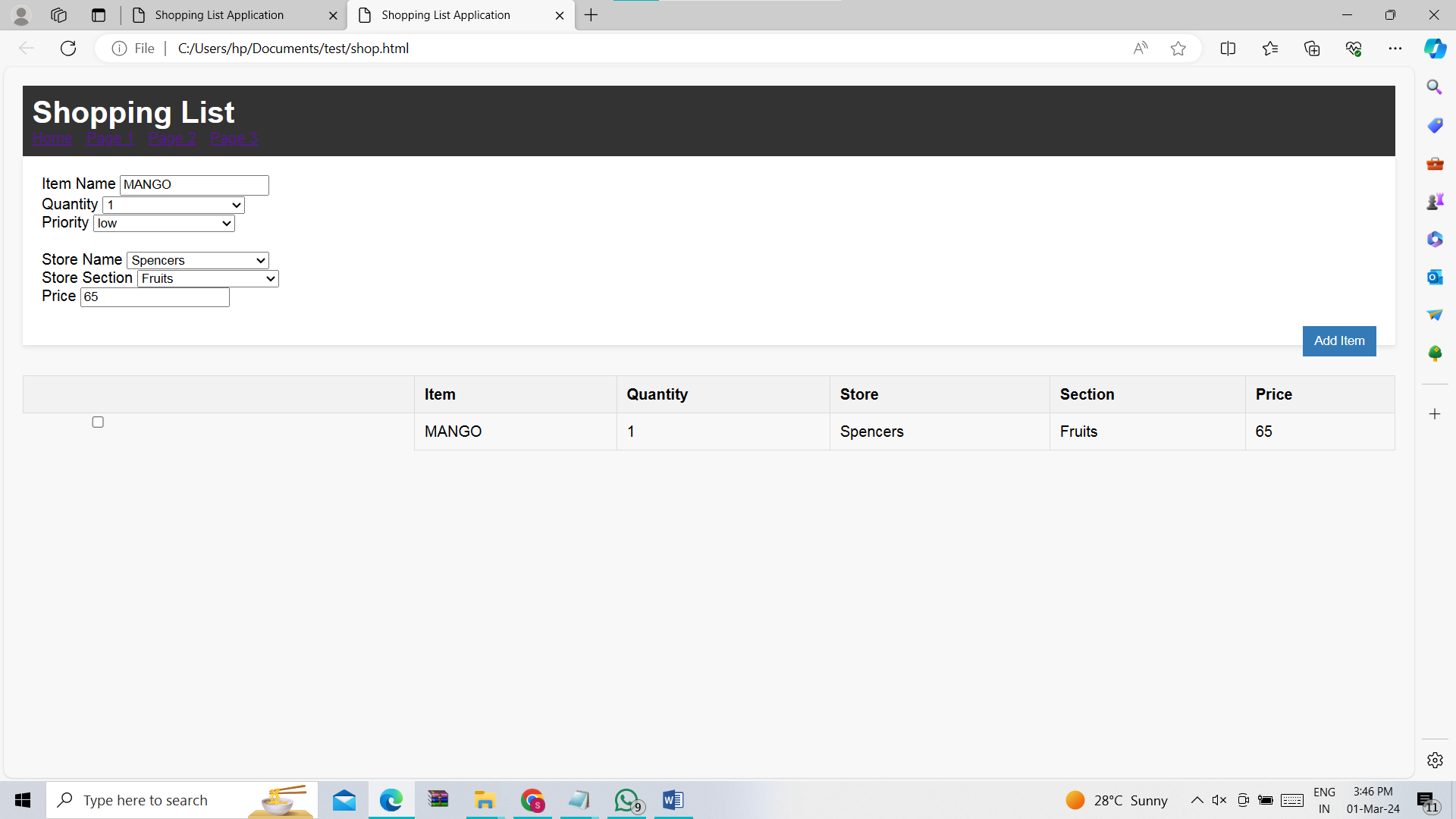
</div>

</div>

<script src="shopping.js" type="text/javascript"></script>

</body>

</html>



Shopping.js-

var stores = ['Fareway', 'Ace Hardware', 'Caseys', 'The Hatchery', 'Amundsens']

var sections = ['Produce', 'Meats', 'Cereal', 'Canned Goods', 'Frozen Foods', 'Dairy', 'Liquor', 'Tools', 'Clothing']

function clickedon() {

let rowcolids = ['itemname', 'qty', 'store', 'category', 'price']

let vals = []

for (let cid of rowcolids) {

vals.push(document.getElementById(cid).value)

}

makeRow(vals, document.getElementById('shoppinglist'))

}

function makeRow(valueList, parent) {

let row = document.createElement("tr")

row.classList.add(document.getElementById("priority").value)

let cb = document.createElement("input")

cb.type = "checkbox"

cb.classList.add("form-control")

row.appendChild(cb)

for (let val of valueList) {

let td = document.createElement("td")

td.innerHTML = val

row.appendChild(td)

}

parent.appendChild(row)

}

function populateSelect(selectId, sList) {

let sel = document.getElementById(selectId)

for (let s of sList) {

let opt = document.createElement("option")

opt.value = s

opt.innerHTML = s

sel.appendChild(opt)

}

}

$(document).ready(function () {

populateSelect('store', stores)

populateSelect('category', sections)

})

Model.js-

// Model

class ShoppingListModel {

constructor() {

this.items = [];

}

addItem(itemName) {

this.items.push(itemName);

}

removeItem(index) {

this.items.splice(index, 1);

}

}

View.js-

// View

class ShoppingListView {

constructor(controller) {

this.controller = controller;

this.itemInput = document.getElementById('itemInput');

this.shoppingList = document.getElementById('shoppingList');

}

bindAddItem(handler) {

this.itemInput.addEventListener('keyup', function (event) {

if (event.key === 'Enter') {

handler(this.itemInput.value.trim());

}

});

document.getElementById('addButton').addEventListener('click', () => {

handler(this.itemInput.value.trim());

});

}

bindRemoveItem(handler) {

this.shoppingList.addEventListener('click', (event) => {

if (event.target.tagName === 'BUTTON') {

const index = Array.from(event.target.parentNode.parentNode.children).indexOf(event.target.parentNode);

handler(index);

}

});

}

clearInput() {

this.itemInput.value = '';

}

render(items) {

this.shoppingList.innerHTML = items

.map((item, index) => `

<li>

<span>${item}</span>

<button onclick="removeItem(this)">Remove</button>

</li>

`)

.join('');

}

}

Controller.js-

// Controller

class ShoppingListController {

constructor(model, view) {

this.model = model;

this.view = view;

// Bind methods

this.handleAddItem = this.handleAddItem.bind(this);

this.handleRemoveItem = this.handleRemoveItem.bind(this);

// Subscribe to view events

this.view.bindAddItem(this.handleAddItem);

this.view.bindRemoveItem(this.handleRemoveItem);

}

handleAddItem(itemName) {

if (itemName !== '') {

this.model.addItem(itemName);

this.view.clearInput();

this.view.render(this.model.items);

}

}

handleRemoveItem(index) {

this.model.removeItem(index);

this.view.render(this.model.items);

}

}

// Initialize the application

const model = new ShoppingListModel();

const view = new ShoppingListView();

const controller = new ShoppingListController(model, view);