

=> shopping.html

```
<html>
  <head>

    <title>Shopping List Application</title>

    <link rel="stylesheet" href="shopping.css">

  </head>

  <body>
    <div class="container">
      <nav class="navbar navbar-default">
        <div class="container-fluid">
          <div class="navbar-header">
            <a class="navbar-brand" href="#">Shopping
List</a>

          </div>
          <ul class="nav navbar-nav">
            <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>
        </div>
      </nav>
      <div class="panel panel-default col-md-12">
        <div class="panel-body">
          <div class="row form-inline">

            <div class="form-group">
              <label for="itemname">Item Name</label>
              <input type="text" class="form-control" name="itemname"
id=itemname />
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        <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="row form-inline" style="margin-top: 20px;
margin-bottom: 20px;">
            <div class="form-group">
            <label for="store">Store Name</label>
            <select name="store" id="store" class="form-control">
            </select>
            </div>
            <div class="form-group">
            <label for="category">Store Section</label>
            <select name="category" id="category"
class="form-control">
            </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-inline" style="float: right;">
    <button onclick="clickedon()" class="btn btn-primary"
id=gobutt>Add Item</button>
</div>
</div>
<div class="col-md-12">
    <table class="table table-bordered">
        <thead>
            <tr>
<th></th><th>Item</th><th>Quantity</th><th>Store</th><th>Section</th><th>P
rice</th>
                </tr>
            </thead>
            <tbody id="shoppinglist"></tbody>
        </table>
    </div>
</div>

    <select id="myid"></select>
<script type="text/javascript">
    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)
        }
    }
    populateSelect('myid', ['one', 'two', 'three', 'four'])
</script>

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

```

```
    <script src="model.js"></script>
    <script src="view.js"></script>
    <script src="congtroller.js"></script>
  </body>
</html>
```

=> shopping.css

```
.nav{
  display: flex;
  justify-content: space-around;
}
.navbar{
  background-color: black;
}
.navbar-header{
  display: flex;
  padding: 10px;
}
a{
  text-decoration: none;
  color: red;
  font-size: 30px;
}
.panel, .panel-default, .col-md-12 {
  border: 2px solid black;
  border-collapse: collapse;
  display: flex;
  flex-direction: column;
  gap: 10px;
  background-color: yellow;
}
.row, .form-inline{
  display: flex;
  flex-direction: column;
  gap: 10px;
}
```

=> 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

```
'use strict';  
class Item {  
    constructor(name, quantity, priority, store, section, price) {  
        this.name = name;  
        this.quantity = quantity;  
        this.priority = priority;  
        this.store = store;  
        this.section = section;  
        this.price = price;  
  
        this._purchased = false;  
  
    }  
  
    get purchased() {  
        return this._purchased;  
    }  
  
    set purchased(nv) {  
        this._purchased = nv;  
        alert(`${this.name} was purchased`)  
    }  
}  
  
class Subject {  
  
    constructor() {  
        this.handlers = []  
    }  
}
```

```

    subscribe(fn) {
        this.handlers.push(fn);
    }

    unsubscribe(fn) {
        this.handlers = this.handlers.filter(
            function(item) {
                if (item !== fn) {
                    return item;
                }
            }
        );
    }

    publish(msg, someobj) {
        var scope = someobj || window;
        for (let fn of this.handlers) {
            fn(scope, msg)
        }
    }
}

class ShoppingList extends Subject {
    constructor() {
        super()
        this.newItems = []
        this.oldItems = [];
    }

    addItem(it) {
        this.newItems.push(it)
        this.publish('newitem', this)
    }
}

```

=> view.js

```

class ShoppingView {
    constructor(model) {

```

```

        // The bind() method creates a new function that, when called, has
        its this keyword set to the provided value.
        model.subscribe(this.redrawList.bind(this))
    }

    redrawList(shoppingList, msg) {
        let tbl = document.getElementById("shoppinglist")
        tbl.innerHTML = ""
        for (let item of shoppingList.newItems) {
            this.addRow(item, tbl)
        }
    }

    addRow(item, parent) {
        let row = document.createElement("tr")
        row.classList.add(item.priority)
        let cb = document.createElement("input")
        cb.type = "checkbox"
        cb.classList.add("form-control")
        cb.onclick = function() { item.purchased = true; }
        row.appendChild(cb)

        for (let val of ['name', 'quantity', 'store', 'section', 'price'])
        {
            let td = document.createElement("td")
            td.innerHTML = item[val]
            row.appendChild(td)
        }
        parent.appendChild(row)
    }
}

```

=> controller.js

```

var stores = ['Fareway', 'Ace Hardware', 'Caseys', 'The Hatchery',
'Amundsens']
var sections = ['Produce', 'Meats', 'Cereal', 'Canned Goods', 'Frozen
Foods', 'Dairy', 'Liquor', 'Tools', 'Clothing']

var shoppingModel = new ShoppingList()
var myView = new ShoppingView(shoppingModel)

```



```

function clickedon() {
    let rowcolids = ['itemname', 'qty', 'store', 'category', 'price',
'priority']
    let vals = {}
    for (let cid of rowcolids) {
        vals[cid] = document.getElementById(cid).value;
    }
    let it = new Item(vals.itemname, vals.qty, vals.priority, vals.store,
vals.category, vals.price)
    shoppingModel.addItem(it)
}

function populateSelect(selectId, sList) {
    let sel = document.getElementById(selectId, sList)
    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)
})

```

OUTPUT

Shopping List

HomePage 1Page 2Page 3

Item Name

meat

Quantity

4

Priority

high

Store Name

Store Section

Price

234

Add Item

	Item	Quantity	Store	Section	Price
<input type="checkbox"/>	egg	4			234
<input type="checkbox"/>	meat	4			234

two