```
 1 /**
 2 * This function will get JSONObject and
 3 * create listview items to show user's
   items.
 4 * @param result Contains all user items
   from database
 5 */
 6 function displayItemsList(result) {
 7     $('#itemsList li').remove(); //clear
   screen
 8
 9     let jsonObject = JSON.parse(result);
10     let myItems = jsonObject.myitems;
11
12     for (let i = 0; i < myItems.length; i
   ++) {
13         let myItemName = myItems[i].
   ITEM_NAME;
14         let myItemCategory = myItems[i].
   ITEM_CATEGORY;
15         let myItemCost = myItems[i].
   ITEM_COST;
16         let myItemDate = myItems[i].
   ITEM_DATE;
17         alert(myItemName, myItemCategory
   , myItemCost, myItemDate);
18         $('#itemsList').append(newItem(
   myItemName, myItemCategory, myItemCost,
   myItemDate, myItems[i].ID)).trigger('
   create'); //add new item
19     }
20     $('#itemsList').listview("refresh");
21 }
```

```
22
23 /**
24 * gets category from selected name in
   Drop Down menu
25 * @param select selected text
26 */
27 function getCategory(select) {
28     category = select.options[select.
   selectedIndex].text;
29 }
30
31 /**
32  * This function sends the user's input
   fields data
33  * to the ViewModel for saving a new item
34  */
35 function fetchItem() {
36     var name = document.getElementById('
   itemname').value;
37     document.getElementById("itemname").
   required = true;
38
39     var cost = document.getElementById('
   cost').value;
40     document.getElementById("cost").
   required = true;
41
42     var date = document.getElementById('
   date').value;
43     document.getElementById("date").
   required = true;
44     clearForm();
45     window.viewModel.addItem(name,
```

```
45 category, cost, date);
46 }
47 /**
48  * This function clears fields from the
   adding page
49  */
50 function clearForm() {
51     $('#itemname').val("");
52     $('#cost').val("");
53 }
54
55 /**
56  * This function calls the viewModel to
   get
57  * all items for the Add Item page
58  */
59 function showItemsUI() {
60     window.viewModel.showItems();
61 }
62
63 /**
64  * This function gets a specific item's
   id (from the listview)
65  * and notify the ViewModel to delete
   that specific item
66  * @param id a specific id item from the
   listview
67  */
68 function deleteItem(id) {
69     $('#' + id).remove();
70     window.viewModel.deleteItemVM(id);
71 }
72
```

```
73 /**
74  * This function generates dynamically a
    new item
75  * element for the listview
76  * with given details from "
   displayItemsList" function
77  * @param name item's name
78  * @param category item's category
79  * @param cost item's cost
80  * @param date item's date
81  * @param id A specific id to attach for
    the new item element
82  * @returns {HTMLLIElement}
83  */
84 function newItem(name, category, cost,
   date, id) {
85 //creating li tag
86     let newItem = document.createElement
   ("li");
87     newItem.setAttribute("data-role", "
   collapsible");
88     newItem.setAttribute("data-collapsed
   -icon", "edit");
89     newItem.setAttribute("data-iconpos"
   , "right");
90     newItem.setAttribute("data-inset", "
   false");
91     newItem.id = id;
92
93 //creating h2 tag
94     let itemTitle = document.
   createElement("h2");
95     let contentTitle = document.
```

```
 95 createTextNode(name);
 96     itemTitle.appendChild(contentTitle);
 97
 98 //creating div tag (to be a wrapper)
 99     let wrapper = document.createElement
    ("div");
100
101 //creating input tag
102     let nameFieldInput = document.
    createElement("input");
103     nameFieldInput.setAttribute("type",
    "text");
104     nameFieldInput.setAttribute("value"
    , name);
105     nameFieldInput.id = "name" + id;
106
107 //creating input tag
108     let categoryFieldInput = document.
    createElement("input");
109     categoryFieldInput.setAttribute("
    type", "text");
110     categoryFieldInput.setAttribute("
    value", category);
111     categoryFieldInput.id = "category"
     + id;
112
113 //creating input tag
114     let costFieldInput = document.
    createElement("input");
115     costFieldInput.setAttribute("type",
    "text");
116     costFieldInput.setAttribute("value"
    , cost);
```

```
117     costFieldInput.id = "cost" + id;
118
119 //creating input tag
120     let dateFieldInput = document.
    createElement("input");
121     dateFieldInput.setAttribute("type",
    "text");
122     dateFieldInput.setAttribute("value"
    , date);
123     dateFieldInput.id = "date" + id;
124
125
126     let deleteBtn = document.
    createElement("a");
127     deleteBtn.setAttribute("data-role",
    "button");
128     let contentDeleteBtn = document.
    createTextNode("Delete");
129     deleteBtn.appendChild(
    contentDeleteBtn);
130
131     deleteBtn.setAttribute("href", "#add
    ");
132
133     deleteBtn.addEventListener("click",
    function () {
134         deleteItem(id);
135     });
136
137 //headers for Items
138     let yourItemNameHeader = document.
    createElement("h3");
139     let contentYourItemNameHeader =
```

```
139 document.createTextNode("Item Name:");
140     yourItemNameHeader.appendChild(
    contentYourItemNameHeader);
141
142     let categoryHeader = document.
    createElement("h3");
143     let contentCategoryHeader = document
    .createTextNode("Category:");
144     categoryHeader.appendChild(
    contentCategoryHeader);
145
146     let costHeader = document.
    createElement("h3");
147     let contentCostHeader = document.
    createTextNode("Cost:");
148     costHeader.appendChild(
    contentCostHeader);
149
150     let dateHeader = document.
    createElement("h3");
151     let contentDateHeader = document.
    createTextNode("Date:");
152     dateHeader.appendChild(
    contentDateHeader);
153
154
155     wrapper.appendChild(
    yourItemNameHeader);
156     wrapper.appendChild(nameFieldInput);
157
158     wrapper.appendChild(categoryHeader);
159     wrapper.appendChild(
    categoryFieldInput);
```

```
160
161     wrapper.appendChild(costHeader);
162     wrapper.appendChild(costFieldInput);
163
164     wrapper.appendChild(dateHeader);
165     wrapper.appendChild(dateFieldInput);
166
167     wrapper.appendChild(deleteBtn);
168
169     newItem.appendChild(itemTitle);
170     newItem.appendChild(wrapper);
171
172     return newItem;
173 }
174
175 /**
176  * This function calls the viewModel to
    get
177  * all items for the View Report page
178  */
179 function getReportItems(){window.
    viewModel.getReport();}
180
181 /**
182  * This function gets all the list of
    items ,and according to
183  * the selected choice of report by
    months from the UI.
184  * The selected items presented on the
    graph in View Report page
185  * @param result Contains all user items
     from database
186  */
```

```
187 function displayReport(result) {
188
189     var yearSelected = parseInt(document
    .getElementById('year').value);
190     var fromMonthSelected = parseInt(
    document.getElementById('fromdate').
    value);
191     var toMonthSelected = parseInt(
    document.getElementById('todate').value
    );
192
193     let jsonObject = JSON.parse(result);
194     let myItems = jsonObject.myitems;
195     var counterArray = [0, 0, 0, 0, 0, 0
    , 0, 0, 0, 0, 0, 0, 0];
196
197     for (let i = 0; i < myItems.length;
    i++) {
198
199         var myItemCategory = myItems[i].
    ITEM_CATEGORY;
200         var myItemCost = parseInt(
    myItems[i].ITEM_COST);
201         var myItemDate = myItems[i].
    ITEM_DATE;
202
203         var res = myItemDate.split("-");
204         var monthDB = parseInt(res[1]);
205         var yearDB = parseInt(res[0]);
206
207         if (monthDB >= fromMonthSelected
     && monthDB <= toMonthSelected && yearDB
     == yearSelected) {
```

```
208             if (myItemCategory == "Food"
    )
209                 counterArray[0] =
    counterArray[0] + myItemCost;
210             else if (myItemCategory == "
    Furnitures")
211                 counterArray[1] =
    counterArray[1] + myItemCost;
212             else if (myItemCategory == "
    Home Tax")
213                 counterArray[2] =
    counterArray[2] + myItemCost;
214             else if (myItemCategory == "
    Mortgage")
215                 counterArray[3] =
    counterArray[3] + myItemCost;
216             else if (myItemCategory == "
    Fuel")
217                 counterArray[4] =
    counterArray[4] + myItemCost;
218             else if (myItemCategory == "
    Car Fix")
219                 counterArray[5] =
    counterArray[5] + myItemCost;
220             else if (myItemCategory == "
    Water")
221                 counterArray[6] =
    counterArray[6] + myItemCost;
222             else if (myItemCategory == "
    Electric")
223                 counterArray[7] =
    counterArray[7] + myItemCost;
224             else if (myItemCategory == "
```

```
224 Phone")
225                     counterArray[8] =
    counterArray[8] + myItemCost;
226                 else if (myItemCategory == "
    Internet")
227                     counterArray[9] =
    counterArray[9] + myItemCost;
228                 else if (myItemCategory == "
    Gas")
229                     counterArray[10] =
    counterArray[10] + myItemCost;
230                 else if (myItemCategory == "
    Closes")
231                     counterArray[11] =
    counterArray[11] + myItemCost;
232                 else if (myItemCategory == "
    Etc")
233                     counterArray[12] =
    counterArray[12] + myItemCost;
234             }
235         }
236         /**
237         * Creates the graph on the UI
238         */
239         am4core.ready(function () {
240
241 // Themes begin
242             am4core.useTheme(
    am4themes_animated);
243 // Themes end
244
245             var chart = am4core.create("
    chartdiv", am4charts.SlicedChart);
```

```
246         chart.hiddenState.properties.
    opacity = 0; // this makes initial fade
    in effect
247
248         chart.data = [{
249             "name": "Food",
250             "value": counterArray[0]
251         }, {
252             "name": "Furnitures",
253             "value": counterArray[1]
254         }, {
255             "name": "Home Tax",
256             "value": counterArray[2]
257         }, {
258             "name": "Mortgage",
259             "value": counterArray[3]
260         }, {
261             "name": "Fuel",
262             "value": counterArray[4]
263         }, {
264             "name": "Car Fix",
265             "value": counterArray[5]
266         }, {
267             "name": "Water",
268             "value": counterArray[6]
269         }, {
270             "name": "Electric",
271             "value": counterArray[7]
272         }, {
273             "name": "Phone",
274             "value": counterArray[8]
275         }, {
276             "name": "Internet",
```

```
277                 "value": counterArray[9]
278             }, {
279                 "name": "Gas",
280                 "value": counterArray[10]
281             }, {
282                 "name": "Closes",
283                 "value": counterArray[11]
284             }, {
285                 "name": "Etc",
286                 "value": counterArray[12]
287             }];
288 
289             var series = chart.series.push(
    new am4charts.FunnelSeries());
290             series.colors.step = 2;
291             series.dataFields.value = "value
    ";
292             series.dataFields.category = "
    name";
293             series.alignLabels = true;
294 
295             series.labelsContainer.
    paddingLeft = -5;
296             series.labelsContainer.width =
    100;
297 
298         }); // end am4core.ready()
299 }
300 
301 
302 
303 
```