

```

1 // Budget Controller
2
3 var budgetController = (function () {
4
5     var Expense = function (id, description, value) {
6         this.id = id,
7         this.description = description;
8         this.value = value,
9         this.percentage = -1
10    };
11
12    Expense.prototype.calcPercentage = function (totalIncome) {
13        if (totalIncome > 0) {
14            this.percentage = Math.round((this.value / totalIncome) * 100);
15        }
16        else {
17            this.percentage = -1;
18        }
19    };
20
21    Expense.prototype.getPercentage = function () {
22        return this.percentage;
23    };
24
25    var Income = function (id, description, value) {
26        this.id = id,
27        this.description = description,
28        this.value = value;
29    };
30
31    var calculateTotal = function (type) {
32        var sum = 0;
33        data.allItems[type].forEach(function (curr) {
34            sum += curr.value;
35        });
36        data.totals[type] = sum;
37    };
38
39    var data = {
40        allItems: {
41            exp: [],
42            inc: []
43        },
44        totals: {
45            exp: 0,
46            inc: 0
47        },
48        budget: 0,
49        percentage: -1
50    };
51
52    return {
53        addItem: function (type, des, val) {
54            var newItem, ID;
55

```

```

56 //create new ID
57 //console.log(data);
58 if (data.allItems[type].length > 0) {
59     ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
60 }
61 else {
62     ID = 0;
63 }
64 if (type === 'exp') {
65     newItem = new Expense(ID, des, val);
66 }
67 else {
68     newItem = new Income(ID, des, val);
69 }
70 //push it into our data structure
71 data.allItems[type].push(newItem);
72
73 //return the new element
74 return newItem;
75 },
76
77 deleteItem: function (type, id) {
78     var ids, index;
79     ids = data.allItems[type].map(function (current) {
80         return current.id;
81     });
82     index = ids.indexOf(id);
83     if (index !== -1) {
84         data.allItems[type].splice(index, 1);
85     }
86 },
87
88 calculateBudget: function () {
89     //calculate total expense and income
90     calculateTotal("exp");
91     calculateTotal("inc");
92
93     //calculate the budget: income -expense
94     data.budget = data.totals.inc - data.totals.exp;
95
96     //calculate the percentage of income that we spent
97     if (data.totals.inc > 0) {
98         data.percentage = Math.round((data.totals.exp / data.totals.inc) *
99         100);
100     }
101     else {
102         data.percentage = -1;
103     }
104 },
105
106 calculatePercentage: function () {
107     data.allItems.exp.forEach(function (curr) {
108         curr.calcPercentage(data.totals.inc);
109     });
110 },

```

```

111
112   getPercentage: function () {
113       var allPerc = data.allItems.exp.map(function (curr) {
114           return curr.getPercentage();
115       });
116       return allPerc;
117   },
118
119   getBudget: function () {
120       return {
121           budget: data.budget,
122           totalInc: data.totals.inc,
123           totalExp: data.totals.exp,
124           percentage: data.percentage
125       },
126   },
127
128   testing: function () {
129       console.log(data);
130   }
131 };
132
133 })();
134
135 //UI controller
136
137 var uiController = (function () {
138
139     var domsStrings = {
140         inputType: '.add__type',
141         inputDescription: '.add__description',
142         inputValue: '.add__value',
143         inputBtn: '.add__btn',
144         incomeContainer: '.income__list',
145         expenseContainer: '.expenses__list',
146         budgetLabel: '.budget__value',
147         incomeLabel: '.budget__income--value',
148         expensesLabel: '.budget__expenses--value',
149         percentageLabel: '.budget__expenses--percentage',
150         container: '.container',
151         expensesPercentageLabel: '.item__percentage',
152         monthAndYear: '.budget__title--month'
153     };
154     return {
155         getInput: function () {
156             return {
157                 type: document.querySelector(domsStrings.inputType).value,
158                 description:
159 document.querySelector(domsStrings.inputDescription).value,
159                 value:
160 parseFloat(document.querySelector(domsStrings.inputValue).value)
161             };
162         },
163
164         addListItem: function (obj, type) {
165             //create HTML string with placeholder text

```

```

165       var html, newHtml, element;
166
167       if (type === 'inc') {
168           element = domsStrings.incomeContainer;
169           html = '<div class="item clearfix" id="inc-%id%"><div
170 class="item__description">%description%</div><div class="right clearfix">' +
171               '<div class="item__value">+ %value%</div><div
172 class="item__delete"><button class="item__delete--btn"><i class="ion-ios-close-
173 outline">' +
174               '</i></button></div></div></div>'
175       }
176       else if (type === 'exp') {
177           element = domsStrings.expenseContainer;
178           html = '<div class="item clearfix" id="exp-%id%"><div
179 class="item__description">%description%</div><div class="right clearfix">' +
180               '<div class="item__value">- %value%</div><div
181 class="item__percentage">21%</div><div class="item__delete">' +
182               '<button class="item__delete--btn"><i class="ion-ios-close-
183 outline"></i></button></div></div></div>'
184       }
185       //Replace the placeholder test with some actual data
186
187       newHtml = html.replace('%id%', obj.id);
188       newHtml = newHtml.replace('%description%', obj.description);
189       //console.log(obj.value);
190
191       newHtml = newHtml.replace('%value%', this.formatNumber(obj.value));
192
193       //Insert the Html into the DOM
194
195       document.querySelector(element).insertAdjacentHTML('beforeend',
196 newHtml);
197
198   },
199
200   deleteListItems: function (selectorID) {
201       var el;
202       el = document.getElementById(selectorID);
203       el.parentNode.removeChild(el);
204   },
205
206   clearFields: function () {
207       var fields, fieldsArrays;
208       fields = document.querySelectorAll(domsStrings.inputDescription + ', '
209 + domsStrings.inputValue);
210
211       fieldsArrays = Array.prototype.slice.call(fields);
212
213       //samajh ni aya
214       fieldsArrays.forEach(function (curr, index, array) {
215           curr.value = '';
216       });
217       fieldsArrays[0].focus();
218   },
219
220
221
222
223

```

```

214     displayBudget: function (obj) {
215
216         document.querySelector(domsStrings.budgetLabel).textContent =
217         this.formatNumber(obj.budget);
218         document.querySelector(domsStrings.incomeLabel).textContent =
219         this.formatNumber(obj.totalInc);
220         document.querySelector(domsStrings.expensesLabel).textContent =
221         this.formatNumber(obj.totalExp);
222
223         if (obj.percentage > 0) {
224
225             document.querySelector(domsStrings.percentageLabel).textContent =
226             obj.percentage + '%';
227
228         }
229         else {
230             document.querySelector(domsStrings.percentageLabel).textContent =
231             '--';
232         }
233     },
234     displayPercentage: function (percentage) {
235
236         var fields =
237         document.querySelectorAll(domsStrings.expensesPercentageLabel);
238         //console.log(fields);
239
240         var nodeListForEach = function (list, callback) {
241             for (var i = 0; i < list.length; i++) {
242                 callback(list[i], i);
243             }
244         };
245
246         nodeListForEach(fields, function (current, index) {
247
248             if (percentage[index] > 0) {
249                 current.textContent = percentage[index] + '%';
250             }
251             else {
252                 current.textContent = '--';
253             }
254         });
255     },
256     formatNumber: function (num) {
257         if (num !== 0 && num !== undefined) {
258             var newNum = num.toFixed(2);
259             var numObj = newNum.split('.');
260             //console.log(numObj);
261             numObj[0] = parseInt(numObj[0]);
262             numObj[0] = numObj[0].toLocaleString();
263             numObj[1] = parseInt(numObj[1]);
264             //console.log(numObj);
265             return (numObj[0] + '.' + numObj[1]);
266         }
267         else if (num === undefined) {

```

```

265         return 0.00;
266     }
267     else {
268         return 0.00;
269     }
270 },
271 displayMonth: function () {
272
273     var now, year, month;
274     now = new Date();
275     year = now.getFullYear();
276     month = now.toLocaleString('default', { month: 'long' });
277
278     document.querySelector(domsStrings.monthAndYear).textContent = month +
279     ', ' + year;
280 },
281 getDomStrings: function () {
282     return domsStrings;
283 }
284 };
285
286 })();
287
288 // Global App Controller
289
290 var controller = (function (budgetCtrl, uiCtrl) {
291
292     var setupEventListeners = function () {
293
294         var DomStrings = uiCtrl.getDomStrings();
295         //console.log(DomStrings);
296
297         document.querySelector(DomStrings.inputBtn).addEventListener('click',
298         ctrlAddItem);
299
300         document.addEventListener('keypress', function (event) {
301             if (event.keyCode === 13 || event.which === 12) {
302                 ctrlAddItem();
303             }
304         });
305
306         document.querySelector(DomStrings.container).addEventListener('click',
307         ctrlDeleteItem);
308
309     };
310
311     var updateBudget = function () {
312
313         //1. calculate the budget
314         budgetCtrl.calculateBudget();
315
316         //2. return the budget
317         var budget = budgetCtrl.getBudget();
318         //console.log(budget);
319         //3. Display the budget on the UI

```

```

318     uiCtrl.displaybudget(budget);
319
320 };
321
322 var updatePercentage = function () {
323     //1. calculate percentages
324     budgetCtrl.calculatePercentage();
325
326     //2. Read percentages from the budget controller
327     var percentages = budgetCtrl.getPercentage();
328     //console.log(percentages);
329
330     //3. Update the UI with the new percentage
331     uiCtrl.displayPercentage(percentages);
332
333 }
334
335 var ctrlAddItem = function () {
336     //console.log('button clicked');
337
338     var newItem, input;
339
340     //1.get the filled input data
341     var input = uiCtrl.getInput();
342     //console.log(input);
343     if (input.description !== "" && !isNaN(input.value) && input.value > 0) {
344
345         //2. add the item to the budget controller
346         newItem = budgetCtrl.addItem(input.type, input.description,
input.value);
347
348         //3. add the item to the UI
349
350         uiCtrl.addListItem(newItem, input.type);
351
352         //4. clear fields
353         uiCtrl.clearFields();
354
355         //5. calculate and update budget
356         updateBudget();
357
358         //6. calculate and update percentages
359         updatePercentage();
360     }
361
362 };
363
364 var ctrlDeleteItem = function (event) {
365     var ItemID, splitID, type, id;
366     ItemID = event.target.parentNode.parentNode.parentNode.id;
367     //console.log(ItemID);
368     if (ItemID) {
369         //inc-1
370         splitID = ItemID.split('-');
371         //console.log(splitID);
372         type = splitID[0];

```

```

373     id = parseInt(splitID[1]);
374
375     //1. delete the item from the data structure
376     budgetCtrl.delelteItem(type, id);
377
378     //2. delete the item from the UI
379     uiCtrl.deletelistItems(ItemID);
380
381     //3. update and show the new budget
382     updateBudget();
383
384     updatePercentage();
385 }
386 };
387
388 return {
389     init: function () {
390         console.log('App has started');
391         uiCtrl.displaybudget({
392             budget: 0,
393             totalInc: 0,
394             toatlExp: 0,
395             percentage: 0
396         });
397         setupEventListeners();
398         uiCtrl.displayMonth();
399     }
400 }
401
402 })(budgetController, uiController);
403
404 controller.init();

```