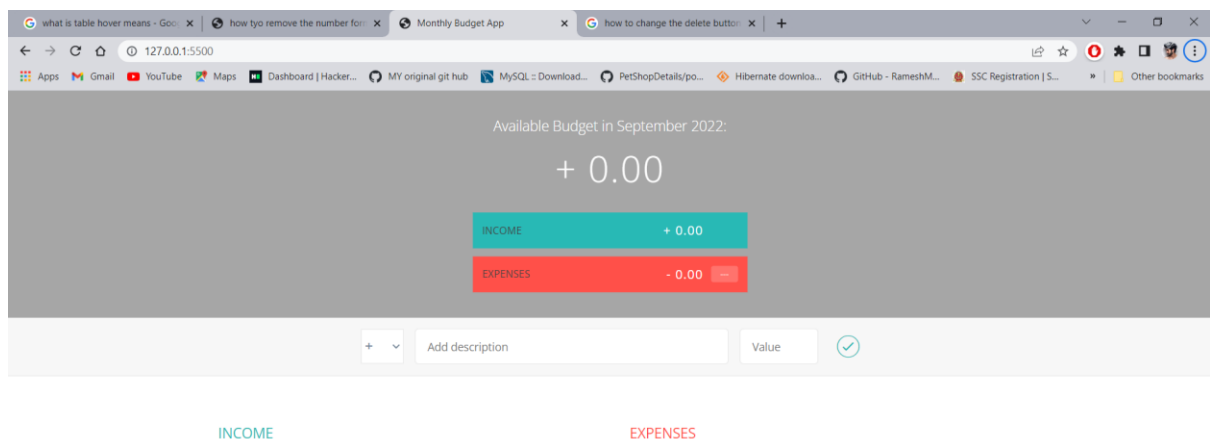


Budget App

- Created 1.Index.html in Vss
- Gave the 2.Style.css link in Index
- Gave the 3.app.js inside script tag of index

Here is our output Webpage.

We can add the income & expense .



what is table hover means - Google | how to remove the number for... | Monthly Budget App | how to change the delete button... | +

127.0.0.1:5500

Available Budget in September 2022:

+ 4,500.00

INCOME	+ 4,500.00
EXPENSES	- 0.00

+ Add description Value ✓

INCOME	EXPENSES
My Salary	+ 4,500.00

what is table hover means - Google | how to remove the number for... | Monthly Budget App | how to change the delete button... | +

127.0.0.1:5500

Available Budget in September 2022:

+ 4,500.00

INCOME	+ 4,500.00
EXPENSES	- 0.00

+ Loan Sanctioned this Month 2500 ✓

INCOME	EXPENSES
My Salary	+ 4,500.00

what is table hover means - Google x how to remove the number for x Monthly Budget App x how to change the delete button x +

127.0.0.1:5500

Available Budget in September 2022:

+ 6,450.00

INCOME + 7,000.00

EXPENSES - 550.00 8%

- EMI's Cutoff 790 ✓

INCOME		EXPENSES	
My Salary	+ 4,500.00	Travelling Expenses	- 550.00 8%
Loan Sanctioned this Month	+ 2,500.00		

what is table hover means - Google x how to remove the number for x Monthly Budget App x how to change the delete button x +

127.0.0.1:5500

Available Budget in September 2022:

+ 1,370.00

INCOME + 7,700.00

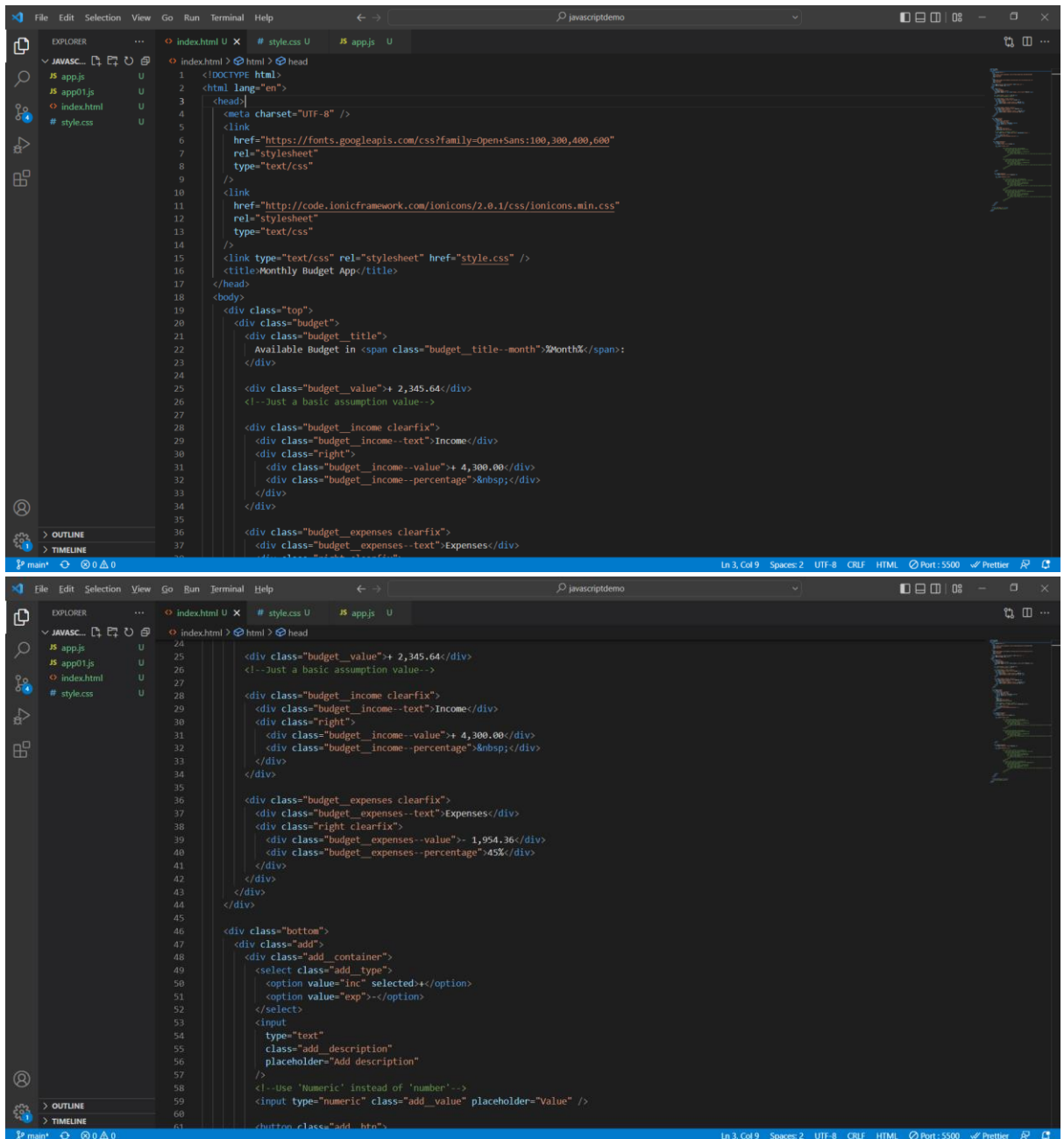
EXPENSES - 6,330.00 52%

+ Add description Value ✓

INCOME		EXPENSES	
My Salary	+ 4,500.00	Travelling Expenses	- 550.00 7%
Loan Sanctioned this Month	+ 2,500.00	EMI's Cutoff	- 790.00 10%
Pension	+ 700.00	Medical Expenses	- 950.00 12%
		Food & Other Expenses	- 2,540.00 33%
		Recharge&Subscriptions	- 1,500.00 19%

By using Cross-button When hovered over the list, We can delete the expense or income from the list.

1. Here I attached the index.html code screen shots

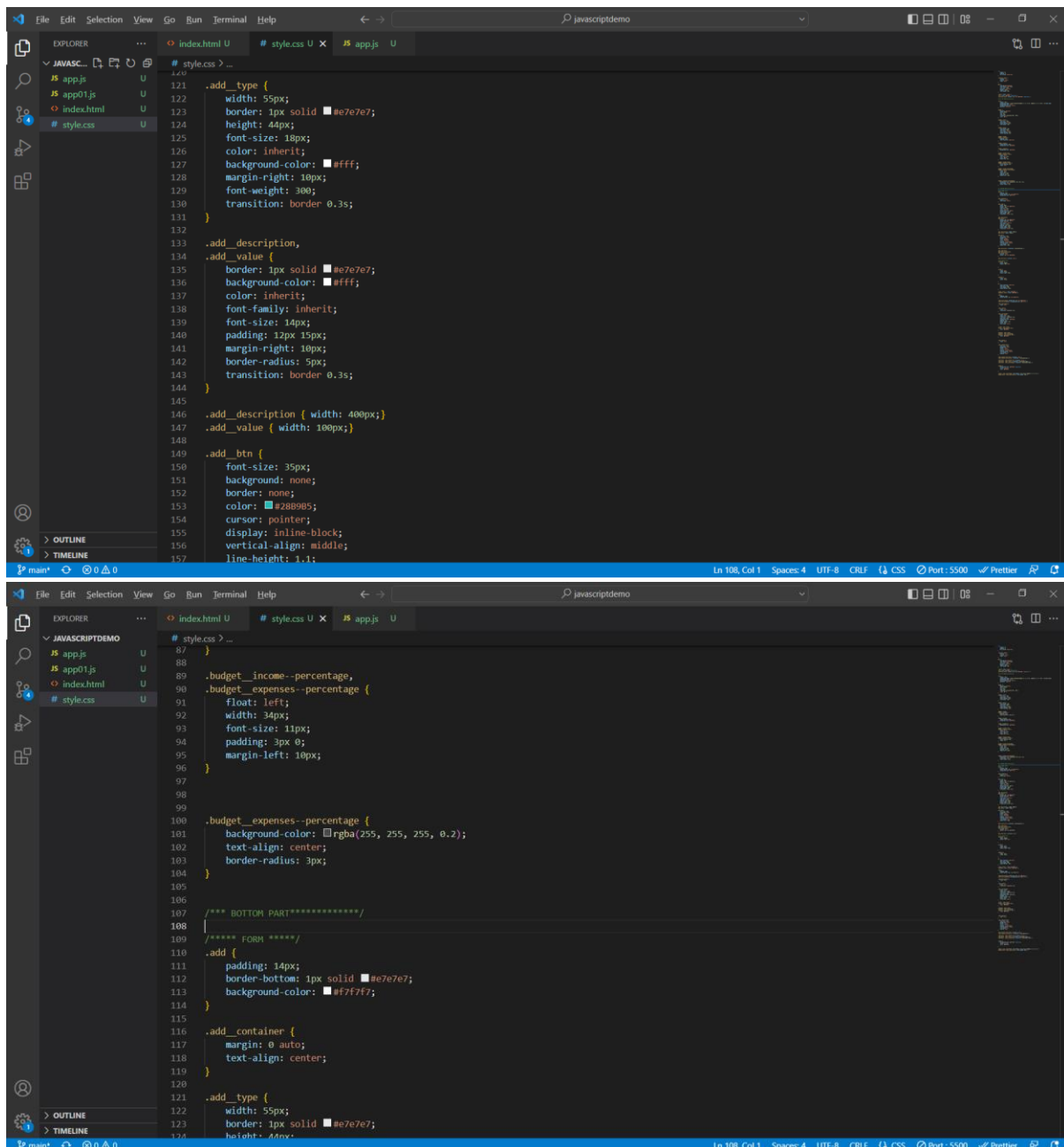


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <link
6     href="https://fonts.googleapis.com/css?family=Open+Sans:100,300,400,600"
7     rel="stylesheet"
8     type="text/css"
9   />
10  <link
11    href="http://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css"
12    rel="stylesheet"
13    type="text/css"
14  />
15  <link type="text/css" rel="stylesheet" href="style.css" />
16  <title>Monthly Budget App</title>
17 </head>
18 <body>
19   <div class="top">
20     <div class="budget">
21       <div class="budget_title">
22         Available Budget in <span class="budget_title-month">Month</span>:
23       </div>
24       <div class="budget_value">+ 2,345.64</div>
25       <!-- Just a basic assumption value -->
26       <div class="budget_income clearfix">
27         <div class="budget_income--text">Income</div>
28         <div class="right">
29           <div class="budget_income--value">+ 4,300.00</div>
30           <div class="budget_income--percentage">&nbsp;&nbsp;&nbsp;</div>
31         </div>
32       </div>
33       <div class="budget_expenses clearfix">
34         <div class="budget_expenses--text">Expenses</div>
35         <div class="right clearfix">
36           <div class="budget_expenses--value">- 1,954.36</div>
37           <div class="budget_expenses--percentage">45%</div>
38         </div>
39       </div>
40     </div>
41     <div class="bottom">
42       <div class="add">
43         <div class="add_container">
44           <select class="add_type">
45             <option value="inc" selected>+</option>
46             <option value="exp">-</option>
47           </select>
48           <input
49             type="text"
50             class="add_description"
51             placeholder="Add description"
52           />
53           <!-- Use 'Numeric' instead of 'number' -->
54           <input type="numeric" class="add_value" placeholder="Value" />
55           <button class="add_btn">Add</button>
56         </div>
57       </div>
58     </div>
59   </div>
60 </body>
61 </html>
```

```
File Edit Selection View Go Run Terminal Help javascriptdemo
EXPLORER index.html U # style.css U JS app.js U
JAVASCRIPT index.html U
JS app.js U
JS app01.js U
index.html U
# style.css U
58 <!--Use 'Numeric' instead of 'number'-->
59 <input type="numeric" class="add_value" placeholder="Value" />
60
61 <button class="add_btn">
62   <i class="ion-ios-checkmark-outline"></i>
63 </button>
64 </div>
65 </div>
66
67 <div class="container">
68   <div class="income">
69     <h2 class="income_title">Income</h2>
70
71     <div class="income_list">
72       <!--
73       <div class="item clearfix" id="income-0">
74         <div class="item_description">Salary</div>
75         <div class="right clearfix">
76           <div class="item_value">+ 2,100.00</div>
77           <div class="item_delete">
78             <button class="item_delete_btn"><i class="ion-ios-close-outline"></i></button>
79           </div>
80         </div>
81       </div>
82
83       <div class="item clearfix" id="income-1">
84         <div class="item_description">Sold car</div>
85         <div class="right clearfix">
86           <div class="item_value">+ 1,500.00</div>
87           <div class="item_delete">
88             <button class="item_delete_btn"><i class="ion-ios-close-outline"></i></button>
89           </div>
90         </div>
91       </div>
92     </div>
93   </div>
94 </div>
```

```
File Edit Selection View Go Run Terminal Help javascriptdemo
EXPLORER index.html U # style.css U JS app.js U
JAVASCRIPT index.html U
JS app.js U
JS app01.js U
index.html U
# style.css U
94 </div>
95
96 <div class="expenses">
97   <h2 class="expenses_title">Expenses</h2>
98
99   <div class="expenses_list">
100     <!--
101     <div class="item clearfix" id="expense-0">
102       <div class="item_description">Apartment rent</div>
103       <div class="right clearfix">
104         <div class="item_value">- 900.00</div>
105         <div class="item_percentage">21%</div>
106         <div class="item_delete">
107           <button class="item_delete_btn"><i class="ion-ios-close-outline"></i></button>
108         </div>
109       </div>
110     </div>
111
112     <div class="item clearfix" id="expense-1">
113       <div class="item_description">Grocery shopping</div>
114       <div class="right clearfix">
115         <div class="item_value">- 435.28</div>
116         <div class="item_percentage">10%</div>
117         <div class="item_delete">
118           <button class="item_delete_btn"><i class="ion-ios-close-outline"></i></button>
119         </div>
120       </div>
121     </div>
122   </div>
123 </div>
124 </div>
125 </div>
126
127 <script src="app.js"></script>
128 <!--JavaScript Application-->
129 </body>
130 </html>
```

Here I added only few screenshots of style.css code Because File Exceeding limit . So, I added (full code) the source code in another pdf.



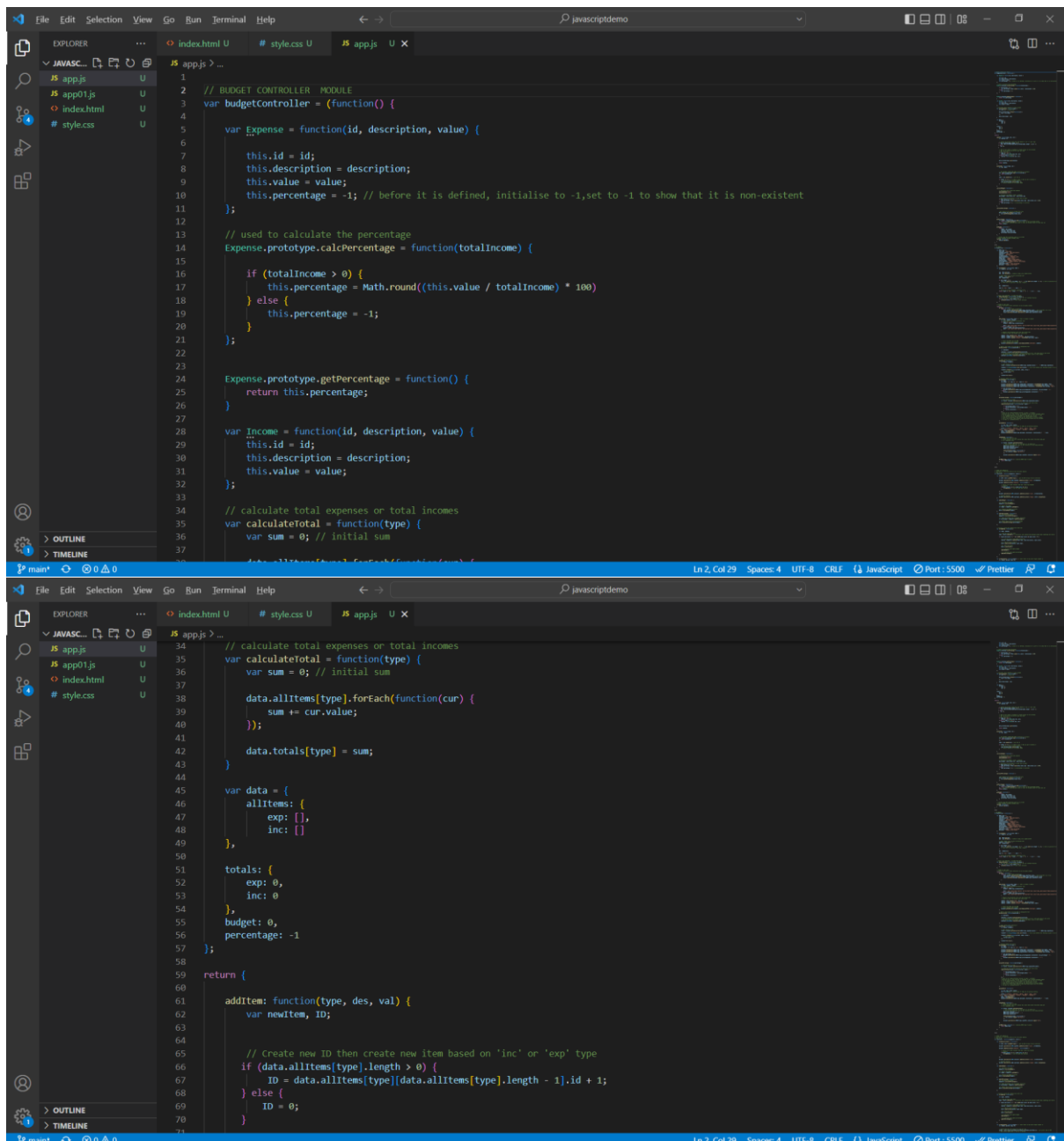
```
File Edit Selection View Go Run Terminal Help javascriptdemo
EXPLORER
  JAVASCRIPTDEMO
    app.js U
    app01.js U
    index.html U
    style.css U
  style.css
    57 letter-spacing: 4px;
    58 }
    59
    60 .budget_income,
    61 .budget_expenses {
    62   padding: 12px;
    63   text-transform: uppercase;
    64 }
    65
    66 .budget_income {
    67   margin-bottom: 10px;
    68   background-color: #288985;
    69 }
    70
    71 .budget_expenses {
    72   background-color: #FF5049;
    73 }
    74
    75 .budget_income-text,
    76 .budget_expenses-text {
    77   float: left;
    78   font-size: 13px;
    79   color: #4444;
    80   margin-top: 2px;
    81 }
    82
    83 .budget_income-value,
    84 .budget_expenses-value {
    85   letter-spacing: 1px;
    86   float: left;
    87 }
    88
    89 .budget_income-percentage,
    90 .budget_expenses-percentage {
    91   float: left;
    92   width: 34px;
    93   font-size: 11px;
    94   padding: 3px 0;
```

```
File Edit Selection View Go Run Terminal Help javascriptdemo
EXPLORER
  JAVASCRIPTDEMO
    app.js U
    app01.js U
    index.html U
    style.css U
  style.css
    24 .red-focus:focus { border: 1px solid #FF5049 !important; }
    25
    26 /**** TOP PART*****/
    27
    28 .top {
    29   height: 40vh;
    30   background-image: linear-gradient(rgba(0, 0, 0, 0.35), rgba(0, 0, 0, 0.35)), url(back.png);
    31   background-size: cover;
    32   background-position: center;
    33   position: relative;
    34 }
    35
    36 .budget {
    37   position: absolute;
    38   width: 350px;
    39   top: 50%;
    40   left: 50%;
    41   transform: translate(-50%, -50%);
    42   color: #fff;
    43 }
    44
    45 .budget_title {
    46   font-size: 18px;
    47   text-align: center;
    48   margin-bottom: 10px;
    49   font-weight: 300;
    50 }
    51
    52 .budget_value {
    53   font-weight: 300;
    54   font-size: 46px;
    55   text-align: center;
    56   margin-bottom: 25px;
    57   letter-spacing: 2px;
    58 }
    59
    60 .budget_income,
    61 .budget_expenses {
```

```
1 * {
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box;
5 }
6
7 .clearfix:after {
8   content: "";
9   display: table;
10  clear: both;
11 }
12
13
14 body {
15   color: lightslategray;
16   font-family: open sans;
17   font-size: 16px;
18   position: relative;
19   height: 100vh;
20   font-weight: 400;
21 }
22
23 .right { float: right; }
24 .red { color: #FF5049 !important; }
25 .red-focus:focus { border: 1px solid #FF5049 !important; }
26
27 /**** TOP PART*****/
28 {
29   height: 40vh;
30   background-image: linear-gradient(rgba(0, 0, 0, 0.35), rgba(0, 0, 0, 0.35)), url(back.png);
31   background-size: cover;
32   background-position: center;
33   position: relative;
34 }
35
36 .budget {
37   position: absolute;
```

```
147 .add_value { width: 100px; }
148
149 .add_btn {
150   font-size: 35px;
151   background: none;
152   border: none;
153   color: #28B985;
154   cursor: pointer;
155   display: inline-block;
156   vertical-align: middle;
157   line-height: 1.1;
158   margin-left: 10px;
159 }
160
161 .add_btn:active { transform: translateY(2px); }
162
163 .add_type:focus,
164 .add_description:focus,
165 .add_value:focus {
166   outline: none;
167   border: 1px solid #28B985;
168 }
169
170 .add_btn:focus { outline: none; }
171
172 /**** LISTS ****/
173 .container {
174   width: 1000px;
175   margin: 60px auto;
176 }
177
178
179
180 .income {
181   float: left;
182   width: 475px;
183   margin-right: 50px;
```


Here I added few Screenshots of app.js(javascript code) Due to exceeding file size limit . So I added the full code in another pdf.



The image displays two screenshots of a VS Code editor window, showing the code for a budget controller module in app.js. The editor interface includes a sidebar with a file explorer showing the project structure (app.js, app01.js, index.html, style.css) and a main editor area with the code.

Top Screenshot: Shows the initial part of the code, including the module definition and the Expense constructor function.

```
1 // BUDGET CONTROLLER MODULE
2 var budgetController = (function() {
3
4     var Expense = function(id, description, value) {
5
6         this.id = id;
7         this.description = description;
8         this.value = value;
9         this.percentage = -1; // before it is defined, initialise to -1, set to -1 to show that it is non-existent
10    };
11
12    // used to calculate the percentage
13    Expense.prototype.calcPercentage = function(totalIncome) {
14
15        if (totalIncome > 0) {
16            this.percentage = Math.round((this.value / totalIncome) * 100);
17        } else {
18            this.percentage = -1;
19        }
20    };
21
22    Expense.prototype.getPercentage = function() {
23        return this.percentage;
24    };
25
26    var Income = function(id, description, value) {
27        this.id = id;
28        this.description = description;
29        this.value = value;
30    };
31
32    // calculate total expenses or total incomes
33    var calculateTotal = function(type) {
34        var sum = 0; // initial sum
35
36        data.allItems[type].forEach(function(cur) {
37            sum += cur.value;
38        });
39
40        data.totals[type] = sum;
41    }
42
43    var data = {
44        allItems: {
45            exp: [],
46            inc: []
47        },
48        totals: {
49            exp: 0,
50            inc: 0
51        },
52        budget: 0,
53        percentage: -1
54    };
55
56    return {
57
58        addItem: function(type, des, val) {
59            var newItem, ID;
60
61            // Create new ID then create new item based on 'inc' or 'exp' type
62            if (data.allItems[type].length > 0) {
63                ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
64            } else {
65                ID = 0;
66            }
67
68            // Create new item
69            newItem = {
70                id: ID,
71                description: des,
72                value: val,
73                type: type
74            };
75
76            data.allItems[type].push(newItem);
77
78            calculateTotal(type);
79
80            return newItem;
81        }
82    };
83
84    // calculate total expenses or total incomes
85    var calculateTotal = function(type) {
86        var sum = 0; // initial sum
87
88        data.allItems[type].forEach(function(cur) {
89            sum += cur.value;
90        });
91
92        data.totals[type] = sum;
93    }
94
95    var data = {
96        allItems: {
97            exp: [],
98            inc: []
99        },
100       totals: {
101           exp: 0,
102           inc: 0
103       },
104       budget: 0,
105       percentage: -1
106   };
107
108   return {
109
110       addItem: function(type, des, val) {
111           var newItem, ID;
112
113           // Create new ID then create new item based on 'inc' or 'exp' type
114           if (data.allItems[type].length > 0) {
115               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
116           } else {
117               ID = 0;
118           }
119
120           // Create new item
121           newItem = {
122               id: ID,
123               description: des,
124               value: val,
125               type: type
126           };
127
128           data.allItems[type].push(newItem);
129
130           calculateTotal(type);
131
132           return newItem;
133       }
134   };
135
136   // calculate total expenses or total incomes
137   var calculateTotal = function(type) {
138       var sum = 0; // initial sum
139
140       data.allItems[type].forEach(function(cur) {
141           sum += cur.value;
142       });
143
144       data.totals[type] = sum;
145   }
146
147   var data = {
148       allItems: {
149           exp: [],
150           inc: []
151       },
152       totals: {
153           exp: 0,
154           inc: 0
155       },
156       budget: 0,
157       percentage: -1
158   };
159
160   return {
161
162       addItem: function(type, des, val) {
163           var newItem, ID;
164
165           // Create new ID then create new item based on 'inc' or 'exp' type
166           if (data.allItems[type].length > 0) {
167               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
168           } else {
169               ID = 0;
170           }
171
172           // Create new item
173           newItem = {
174               id: ID,
175               description: des,
176               value: val,
177               type: type
178           };
179
180           data.allItems[type].push(newItem);
181
182           calculateTotal(type);
183
184           return newItem;
185       }
186   };
187
188   // calculate total expenses or total incomes
189   var calculateTotal = function(type) {
190       var sum = 0; // initial sum
191
192       data.allItems[type].forEach(function(cur) {
193           sum += cur.value;
194       });
195
196       data.totals[type] = sum;
197   }
198
199   var data = {
200       allItems: {
201           exp: [],
202           inc: []
203       },
204       totals: {
205           exp: 0,
206           inc: 0
207       },
208       budget: 0,
209       percentage: -1
210   };
211
212   return {
213
214       addItem: function(type, des, val) {
215           var newItem, ID;
216
217           // Create new ID then create new item based on 'inc' or 'exp' type
218           if (data.allItems[type].length > 0) {
219               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
220           } else {
221               ID = 0;
222           }
223
224           // Create new item
225           newItem = {
226               id: ID,
227               description: des,
228               value: val,
229               type: type
230           };
231
232           data.allItems[type].push(newItem);
233
234           calculateTotal(type);
235
236           return newItem;
237       }
238   };
239
240   // calculate total expenses or total incomes
241   var calculateTotal = function(type) {
242       var sum = 0; // initial sum
243
244       data.allItems[type].forEach(function(cur) {
245           sum += cur.value;
246       });
247
248       data.totals[type] = sum;
249   }
250
251   var data = {
252       allItems: {
253           exp: [],
254           inc: []
255       },
256       totals: {
257           exp: 0,
258           inc: 0
259       },
260       budget: 0,
261       percentage: -1
262   };
263
264   return {
265
266       addItem: function(type, des, val) {
267           var newItem, ID;
268
269           // Create new ID then create new item based on 'inc' or 'exp' type
270           if (data.allItems[type].length > 0) {
271               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
272           } else {
273               ID = 0;
274           }
275
276           // Create new item
277           newItem = {
278               id: ID,
279               description: des,
280               value: val,
281               type: type
282           };
283
284           data.allItems[type].push(newItem);
285
286           calculateTotal(type);
287
288           return newItem;
289       }
290   };
291
292   // calculate total expenses or total incomes
293   var calculateTotal = function(type) {
294       var sum = 0; // initial sum
295
296       data.allItems[type].forEach(function(cur) {
297           sum += cur.value;
298       });
299
300       data.totals[type] = sum;
301   }
302
303   var data = {
304       allItems: {
305           exp: [],
306           inc: []
307       },
308       totals: {
309           exp: 0,
310           inc: 0
311       },
312       budget: 0,
313       percentage: -1
314   };
315
316   return {
317
318       addItem: function(type, des, val) {
319           var newItem, ID;
320
321           // Create new ID then create new item based on 'inc' or 'exp' type
322           if (data.allItems[type].length > 0) {
323               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
324           } else {
325               ID = 0;
326           }
327
328           // Create new item
329           newItem = {
330               id: ID,
331               description: des,
332               value: val,
333               type: type
334           };
335
336           data.allItems[type].push(newItem);
337
338           calculateTotal(type);
339
340           return newItem;
341       }
342   };
343
344   // calculate total expenses or total incomes
345   var calculateTotal = function(type) {
346       var sum = 0; // initial sum
347
348       data.allItems[type].forEach(function(cur) {
349           sum += cur.value;
350       });
351
352       data.totals[type] = sum;
353   }
354
355   var data = {
356       allItems: {
357           exp: [],
358           inc: []
359       },
360       totals: {
361           exp: 0,
362           inc: 0
363       },
364       budget: 0,
365       percentage: -1
366   };
367
368   return {
369
370       addItem: function(type, des, val) {
371           var newItem, ID;
372
373           // Create new ID then create new item based on 'inc' or 'exp' type
374           if (data.allItems[type].length > 0) {
375               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
376           } else {
377               ID = 0;
378           }
379
380           // Create new item
381           newItem = {
382               id: ID,
383               description: des,
384               value: val,
385               type: type
386           };
387
388           data.allItems[type].push(newItem);
389
390           calculateTotal(type);
391
392           return newItem;
393       }
394   };
395
396   // calculate total expenses or total incomes
397   var calculateTotal = function(type) {
398       var sum = 0; // initial sum
399
400       data.allItems[type].forEach(function(cur) {
401           sum += cur.value;
402       });
403
404       data.totals[type] = sum;
405   }
406
407   var data = {
408       allItems: {
409           exp: [],
410           inc: []
411       },
412       totals: {
413           exp: 0,
414           inc: 0
415       },
416       budget: 0,
417       percentage: -1
418   };
419
420   return {
421
422       addItem: function(type, des, val) {
423           var newItem, ID;
424
425           // Create new ID then create new item based on 'inc' or 'exp' type
426           if (data.allItems[type].length > 0) {
427               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
428           } else {
429               ID = 0;
430           }
431
432           // Create new item
433           newItem = {
434               id: ID,
435               description: des,
436               value: val,
437               type: type
438           };
439
440           data.allItems[type].push(newItem);
441
442           calculateTotal(type);
443
444           return newItem;
445       }
446   };
447
448   // calculate total expenses or total incomes
449   var calculateTotal = function(type) {
450       var sum = 0; // initial sum
451
452       data.allItems[type].forEach(function(cur) {
453           sum += cur.value;
454       });
455
456       data.totals[type] = sum;
457   }
458
459   var data = {
460       allItems: {
461           exp: [],
462           inc: []
463       },
464       totals: {
465           exp: 0,
466           inc: 0
467       },
468       budget: 0,
469       percentage: -1
470   };
471
472   return {
473
474       addItem: function(type, des, val) {
475           var newItem, ID;
476
477           // Create new ID then create new item based on 'inc' or 'exp' type
478           if (data.allItems[type].length > 0) {
479               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
480           } else {
481               ID = 0;
482           }
483
484           // Create new item
485           newItem = {
486               id: ID,
487               description: des,
488               value: val,
489               type: type
490           };
491
492           data.allItems[type].push(newItem);
493
494           calculateTotal(type);
495
496           return newItem;
497       }
498   };
499
500   // calculate total expenses or total incomes
501   var calculateTotal = function(type) {
502       var sum = 0; // initial sum
503
504       data.allItems[type].forEach(function(cur) {
505           sum += cur.value;
506       });
507
508       data.totals[type] = sum;
509   }
510
511   var data = {
512       allItems: {
513           exp: [],
514           inc: []
515       },
516       totals: {
517           exp: 0,
518           inc: 0
519       },
520       budget: 0,
521       percentage: -1
522   };
523
524   return {
525
526       addItem: function(type, des, val) {
527           var newItem, ID;
528
529           // Create new ID then create new item based on 'inc' or 'exp' type
530           if (data.allItems[type].length > 0) {
531               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
532           } else {
533               ID = 0;
534           }
535
536           // Create new item
537           newItem = {
538               id: ID,
539               description: des,
540               value: val,
541               type: type
542           };
543
544           data.allItems[type].push(newItem);
545
546           calculateTotal(type);
547
548           return newItem;
549       }
550   };
551
552   // calculate total expenses or total incomes
553   var calculateTotal = function(type) {
554       var sum = 0; // initial sum
555
556       data.allItems[type].forEach(function(cur) {
557           sum += cur.value;
558       });
559
560       data.totals[type] = sum;
561   }
562
563   var data = {
564       allItems: {
565           exp: [],
566           inc: []
567       },
568       totals: {
569           exp: 0,
570           inc: 0
571       },
572       budget: 0,
573       percentage: -1
574   };
575
576   return {
577
578       addItem: function(type, des, val) {
579           var newItem, ID;
580
581           // Create new ID then create new item based on 'inc' or 'exp' type
582           if (data.allItems[type].length > 0) {
583               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
584           } else {
585               ID = 0;
586           }
587
588           // Create new item
589           newItem = {
590               id: ID,
591               description: des,
592               value: val,
593               type: type
594           };
595
596           data.allItems[type].push(newItem);
597
598           calculateTotal(type);
599
600           return newItem;
601       }
602   };
603
604   // calculate total expenses or total incomes
605   var calculateTotal = function(type) {
606       var sum = 0; // initial sum
607
608       data.allItems[type].forEach(function(cur) {
609           sum += cur.value;
610       });
611
612       data.totals[type] = sum;
613   }
614
615   var data = {
616       allItems: {
617           exp: [],
618           inc: []
619       },
620       totals: {
621           exp: 0,
622           inc: 0
623       },
624       budget: 0,
625       percentage: -1
626   };
627
628   return {
629
630       addItem: function(type, des, val) {
631           var newItem, ID;
632
633           // Create new ID then create new item based on 'inc' or 'exp' type
634           if (data.allItems[type].length > 0) {
635               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
636           } else {
637               ID = 0;
638           }
639
640           // Create new item
641           newItem = {
642               id: ID,
643               description: des,
644               value: val,
645               type: type
646           };
647
648           data.allItems[type].push(newItem);
649
650           calculateTotal(type);
651
652           return newItem;
653       }
654   };
655
656   // calculate total expenses or total incomes
657   var calculateTotal = function(type) {
658       var sum = 0; // initial sum
659
660       data.allItems[type].forEach(function(cur) {
661           sum += cur.value;
662       });
663
664       data.totals[type] = sum;
665   }
666
667   var data = {
668       allItems: {
669           exp: [],
670           inc: []
671       },
672       totals: {
673           exp: 0,
674           inc: 0
675       },
676       budget: 0,
677       percentage: -1
678   };
679
680   return {
681
682       addItem: function(type, des, val) {
683           var newItem, ID;
684
685           // Create new ID then create new item based on 'inc' or 'exp' type
686           if (data.allItems[type].length > 0) {
687               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
688           } else {
689               ID = 0;
690           }
691
692           // Create new item
693           newItem = {
694               id: ID,
695               description: des,
696               value: val,
697               type: type
698           };
699
700           data.allItems[type].push(newItem);
701
702           calculateTotal(type);
703
704           return newItem;
705       }
706   };
707
708   // calculate total expenses or total incomes
709   var calculateTotal = function(type) {
710       var sum = 0; // initial sum
711
712       data.allItems[type].forEach(function(cur) {
713           sum += cur.value;
714       });
715
716       data.totals[type] = sum;
717   }
718
719   var data = {
720       allItems: {
721           exp: [],
722           inc: []
723       },
724       totals: {
725           exp: 0,
726           inc: 0
727       },
728       budget: 0,
729       percentage: -1
730   };
731
732   return {
733
734       addItem: function(type, des, val) {
735           var newItem, ID;
736
737           // Create new ID then create new item based on 'inc' or 'exp' type
738           if (data.allItems[type].length > 0) {
739               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
740           } else {
741               ID = 0;
742           }
743
744           // Create new item
745           newItem = {
746               id: ID,
747               description: des,
748               value: val,
749               type: type
750           };
751
752           data.allItems[type].push(newItem);
753
754           calculateTotal(type);
755
756           return newItem;
757       }
758   };
759
760   // calculate total expenses or total incomes
761   var calculateTotal = function(type) {
762       var sum = 0; // initial sum
763
764       data.allItems[type].forEach(function(cur) {
765           sum += cur.value;
766       });
767
768       data.totals[type] = sum;
769   }
770
771   var data = {
772       allItems: {
773           exp: [],
774           inc: []
775       },
776       totals: {
777           exp: 0,
778           inc: 0
779       },
780       budget: 0,
781       percentage: -1
782   };
783
784   return {
785
786       addItem: function(type, des, val) {
787           var newItem, ID;
788
789           // Create new ID then create new item based on 'inc' or 'exp' type
790           if (data.allItems[type].length > 0) {
791               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
792           } else {
793               ID = 0;
794           }
795
796           // Create new item
797           newItem = {
798               id: ID,
799               description: des,
800               value: val,
801               type: type
802           };
803
804           data.allItems[type].push(newItem);
805
806           calculateTotal(type);
807
808           return newItem;
809       }
810   };
811
812   // calculate total expenses or total incomes
813   var calculateTotal = function(type) {
814       var sum = 0; // initial sum
815
816       data.allItems[type].forEach(function(cur) {
817           sum += cur.value;
818       });
819
820       data.totals[type] = sum;
821   }
822
823   var data = {
824       allItems: {
825           exp: [],
826           inc: []
827       },
828       totals: {
829           exp: 0,
830           inc: 0
831       },
832       budget: 0,
833       percentage: -1
834   };
835
836   return {
837
838       addItem: function(type, des, val) {
839           var newItem, ID;
840
841           // Create new ID then create new item based on 'inc' or 'exp' type
842           if (data.allItems[type].length > 0) {
843               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
844           } else {
845               ID = 0;
846           }
847
848           // Create new item
849           newItem = {
850               id: ID,
851               description: des,
852               value: val,
853               type: type
854           };
855
856           data.allItems[type].push(newItem);
857
858           calculateTotal(type);
859
860           return newItem;
861       }
862   };
863
864   // calculate total expenses or total incomes
865   var calculateTotal = function(type) {
866       var sum = 0; // initial sum
867
868       data.allItems[type].forEach(function(cur) {
869           sum += cur.value;
870       });
871
872       data.totals[type] = sum;
873   }
874
875   var data = {
876       allItems: {
877           exp: [],
878           inc: []
879       },
880       totals: {
881           exp: 0,
882           inc: 0
883       },
884       budget: 0,
885       percentage: -1
886   };
887
888   return {
889
890       addItem: function(type, des, val) {
891           var newItem, ID;
892
893           // Create new ID then create new item based on 'inc' or 'exp' type
894           if (data.allItems[type].length > 0) {
895               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
896           } else {
897               ID = 0;
898           }
899
900           // Create new item
901           newItem = {
902               id: ID,
903               description: des,
904               value: val,
905               type: type
906           };
907
908           data.allItems[type].push(newItem);
909
910           calculateTotal(type);
911
912           return newItem;
913       }
914   };
915
916   // calculate total expenses or total incomes
917   var calculateTotal = function(type) {
918       var sum = 0; // initial sum
919
920       data.allItems[type].forEach(function(cur) {
921           sum += cur.value;
922       });
923
924       data.totals[type] = sum;
925   }
926
927   var data = {
928       allItems: {
929           exp: [],
930           inc: []
931       },
932       totals: {
933           exp: 0,
934           inc: 0
935       },
936       budget: 0,
937       percentage: -1
938   };
939
940   return {
941
942       addItem: function(type, des, val) {
943           var newItem, ID;
944
945           // Create new ID then create new item based on 'inc' or 'exp' type
946           if (data.allItems[type].length > 0) {
947               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
948           } else {
949               ID = 0;
950           }
951
952           // Create new item
953           newItem = {
954               id: ID,
955               description: des,
956               value: val,
957               type: type
958           };
959
960           data.allItems[type].push(newItem);
961
962           calculateTotal(type);
963
964           return newItem;
965       }
966   };
967
968   // calculate total expenses or total incomes
969   var calculateTotal = function(type) {
970       var sum = 0; // initial sum
971
972       data.allItems[type].forEach(function(cur) {
973           sum += cur.value;
974       });
975
976       data.totals[type] = sum;
977   }
978
979   var data = {
980       allItems: {
981           exp: [],
982           inc: []
983       },
984       totals: {
985           exp: 0,
986           inc: 0
987       },
988       budget: 0,
989       percentage: -1
990   };
991
992   return {
993
994       addItem: function(type, des, val) {
995           var newItem, ID;
996
997           // Create new ID then create new item based on 'inc' or 'exp' type
998           if (data.allItems[type].length > 0) {
999               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
1000           } else {
1001               ID = 0;
1002           }
1003
1004           // Create new item
1005           newItem = {
1006               id: ID,
1007               description: des,
1008               value: val,
1009               type: type
1010           };
1011
1012           data.allItems[type].push(newItem);
1013
1014           calculateTotal(type);
1015
1016           return newItem;
1017       }
1018   };
1019
1020   // calculate total expenses or total incomes
1021   var calculateTotal = function(type) {
1022       var sum = 0; // initial sum
1023
1024       data.allItems[type].forEach(function(cur) {
1025           sum += cur.value;
1026       });
1027
1028       data.totals[type] = sum;
1029   }
1030
1031   var data = {
1032       allItems: {
1033           exp: [],
1034           inc: []
1035       },
1036       totals: {
1037           exp: 0,
1038           inc: 0
1039       },
1040       budget: 0,
1041       percentage: -1
1042   };
1043
1044   return {
1045
1046       addItem: function(type, des, val) {
1047           var newItem, ID;
1048
1049           // Create new ID then create new item based on 'inc' or 'exp' type
1050           if (data.allItems[type].length > 0) {
1051               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
1052           } else {
1053               ID = 0;
1054           }
1055
1056           // Create new item
1057           newItem = {
1058               id: ID,
1059               description: des,
1060               value: val,
1061               type: type
1062           };
1063
1064           data.allItems[type].push(newItem);
1065
1066           calculateTotal(type);
1067
1068           return newItem;
1069       }
1070   };
1071
1072   // calculate total expenses or total incomes
1073   var calculateTotal = function(type) {
1074       var sum = 0; // initial sum
1075
1076       data.allItems[type].forEach(function(cur) {
1077           sum += cur.value;
1078       });
1079
1080       data.totals[type] = sum;
1081   }
1082
1083   var data = {
1084       allItems: {
1085           exp: [],
1086           inc: []
1087       },
1088       totals: {
1089           exp: 0,
1090           inc: 0
1091       },
1092       budget: 0,
1093       percentage: -1
1094   };
1095
1096   return {
1097
1098       addItem: function(type, des, val) {
1099           var newItem, ID;
1100
1101           // Create new ID then create new item based on 'inc' or 'exp' type
1102           if (data.allItems[type].length > 0) {
1103               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
1104           } else {
1105               ID = 0;
1106           }
1107
1108           // Create new item
1109           newItem = {
1110               id: ID,
1111               description: des,
1112               value: val,
1113               type: type
1114           };
1115
1116           data.allItems[type].push(newItem);
1117
1118           calculateTotal(type);
1119
1120           return newItem;
1121       }
1122   };
1123
1124   // calculate total expenses or total incomes
1125   var calculateTotal = function(type) {
1126       var sum = 0; // initial sum
1127
1128       data.allItems[type].forEach(function(cur) {
1129           sum += cur.value;
1130       });
1131
1132       data.totals[type] = sum;
1133   }
1134
1135   var data = {
1136       allItems: {
1137           exp: [],
1138           inc: []
1139       },
1140       totals: {
1141           exp: 0,
1142           inc: 0
1143       },
1144       budget: 0,
1145       percentage: -1
1146   };
1147
1148   return {
1149
1150       addItem: function(type, des, val) {
1151           var newItem, ID;
1152
1153           // Create new ID then create new item based on 'inc' or 'exp' type
1154           if (data.allItems[type].length > 0) {
1155               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
1156           } else {
1157               ID = 0;
1158           }
1159
1160           // Create new item
1161           newItem = {
1162               id: ID,
1163               description: des,
1164               value: val,
1165               type: type
1166           };
1167
1168           data.allItems[type].push(newItem);
1169
1170           calculateTotal(type);
1171
1172           return newItem;
1173       }
1174   };
1175
1176   // calculate total expenses or total incomes
1177   var calculateTotal = function(type) {
1178       var sum = 0; // initial sum
1179
1180       data.allItems[type].forEach(function(cur) {
1181           sum += cur.value;
1182       });
1183
1184       data.totals[type] = sum;
1185   }
1186
1187   var data = {
1188       allItems: {
1189           exp: [],
1190           inc: []
1191       },
1192       totals: {
1193           exp: 0,
1194           inc: 0
1195       },
1196       budget: 0,
1197       percentage: -1
1198   };
1199
1200   return {
1201
1202       addItem: function(type, des, val) {
1203           var newItem, ID;
1204
1205           // Create new ID then create new item based on 'inc' or 'exp' type
1206           if (data.allItems[type].length > 0) {
1207               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
1208           } else {
1209               ID = 0;
1210           }
1211
1212           // Create new item
1213           newItem = {
1214               id: ID,
1215               description: des,
1216               value: val,
1217               type: type
1218           };
1219
1220           data.allItems[type].push(newItem);
1221
1222           calculateTotal(type);
1223
1224           return newItem;
1225       }
1226   };
1227
1228   // calculate total expenses or total incomes
1229   var calculateTotal = function(type) {
1230       var sum = 0; // initial sum
1231
1232       data.allItems[type].forEach(function(cur) {
1233           sum += cur.value;
1234       });
1235
1236       data.totals[type] = sum;
1237   }
1238
1239   var data = {
1240       allItems: {
1241           exp: [],
1242           inc: []
1243       },
1244       totals: {
1245           exp: 0,
1246           inc: 0
1247       },
1248       budget: 0,
1249       percentage: -1
1250   };
1251
1252   return {
1253
1254       addItem: function(type, des, val) {
1255           var newItem, ID;
1256
1257           // Create new ID then create new item based on 'inc' or 'exp' type
1258           if (data.allItems[type].length > 0) {
1259               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
1260           } else {
1261               ID = 0;
1262           }
1263
1264           // Create new item
1265           newItem = {
1266               id: ID,
1267               description: des,
1268               value: val,
1269               type: type
1270           };
1271
1272           data.allItems[type].push(newItem);
1273
1274           calculateTotal(type);
1275
1276           return newItem;
1277       }
1278   };
1279
1280   // calculate total expenses or total incomes
1281   var calculateTotal = function(type) {
1282       var sum = 0; // initial sum
1283
1284       data.allItems[type].forEach(function(cur) {
1285           sum += cur.value;
1286       });
1287
1288       data.totals[type] = sum;
1289   }
1290
1291   var data = {
1292       allItems: {
1293           exp: [],
1294           inc: []
1295       },
1296       totals: {
1297           exp: 0,
1298           inc: 0
1299       },
1300       budget: 0,
1301       percentage: -1
1302   };
1303
1304   return {
1305
1306       addItem: function(type, des, val) {
1307           var newItem, ID;
1308
1309           // Create new ID then create new item based on 'inc' or 'exp' type
1310           if (data.allItems[type].length > 0) {
1311               ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
1312           } else {
1313               ID = 0;
1314           }
1315
1316           // Create new item
1317           newItem = {
1318               id: ID,
1319               description: des,
1320               value: val,
1321               type: type
1322           };
1323
1324           data.allItems[type].push(newItem);
1325
1326           calculateTotal(type);
1327
1328           return newItem;
1329       }
1330   };
1331
1332   // calculate total expenses or total incomes
1333   var calculateTotal = function(type) {
1334       var sum = 0; // initial sum
1335
1336       data.allItems[type].forEach(function(cur) {
1337           sum += cur.value;
1338       });
1339
1340       data.totals[type] = sum;
1341   }
1342
1343   var data = {
1344       allItems: {
1345           exp: [],
13
```

This screenshot shows a VS Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'JAVASCRIPTDEMO' with files 'app.js', 'app01.js', 'index.html', and 'style.css'. The code editor is open to 'app.js' and shows the following JavaScript code:

```
71 // That we put either in expense or income arrays for the allItems
72 // How can we specify the ID for each new item?
73 // ID = last ID + 1
74 if (type === 'exp') {
75     newItem = new Expense(ID, des, val);
76 } else if (type === 'inc') {
77     newItem = new Income(ID, des, val);
78 }
79
80 data.allItems[type].push(newItem);
81
82 return newItem;
83
84 },
85
86 deleteItem: function(type, id) {
87     var ids, index;
88
89     // map RETURNS A BRAND NEW ARRAY (different to forEach)
90     ids = data.allItems[type].map(function(current) {
91         return current.id;
92     });
93
94     index = ids.indexOf(id); // get the id
95
96     if (index !== -1) { // if index is not -1 then we want to delete it
97         // splice is used to delete element
98         data.allItems[type].splice(index, 1);
99         // removes elements at the number INDEX
100     }
101
102 },
103
104 calculateBudget: function() {
```

This screenshot shows the same VS Code editor window, but the code editor is now open to 'app01.js'. The code editor shows the following JavaScript code:

```
107 calculateBudget: function() {
108     // calculate total income and expenses
109     calculateTotal('exp');
110     calculateTotal('inc');
111
112     // Calculate the budget: income - expenses
113     data.budget = data.totals.inc - data.totals.exp;
114
115     // Calculate the percentage of income that we spent
116     if (data.totals.inc > 0) {
117         data.percentage = Math.round((data.totals.exp / data.totals.inc) * 100);
118     } else {
119         data.percentage = -1; // no percentage to calculate
120     }
121
122 },
123
124 calculatePercentages: function() {
125
126     // get expense array and go through the array
127     data.allItems.exp.forEach(function(cur) {
128         cur.calcPercentage(data.totals.inc);
129     });
130
131 },
132
133 getPercentages: function() {
134     var allPerc = data.allItems.exp.map(function(cur) {
135         return cur.getPercentage(); // return result of get percentage method
136     }); // map returns something and stores it into the variable while for each does not
137     return allPerc;
138 },
139
140 getBudget: function() {
141     return {
142         budget: data.budget
143     };
144 }
```



```
File Edit Selection View Go Run Terminal Help javascriptdemo
EXPLORER index.html U # style.css U JS app.js U X
JS app.js U 319 displayMonth: function() {
JS app.js U 320 var now, year, month, months;
JS app01.js U 321 now = new Date(); // returns date of today by not passing anything
index.html U 322 months = ['January', 'February', 'March', 'April', 'May', 'June',
# style.css U 323 'July', 'August', 'September', 'October', 'November', 'December'];
324 year = now.getFullYear();
325 month = now.getMonth();
326 document.querySelector(DOMstrings.dateLabel).textContent = months[month] + ' ' + year;
327
328 },
329
330 changedType: function() {
331 // change CSS class for this
332 // select elements that will receive this class then select the button and give
333 // it a red/green class
334
335 var fields = document.querySelectorAll(
336 // this returns node list. To loop over it, we cannot use
337 // the for each method but can use the function we wrote nodeListForEach
338 DOMstrings.inputType + ',' +
339 DOMstrings.inputDescription + ',' +
340 DOMstrings.inputValue);
341
342 nodeListForEach(fields, function(cur) {
343 cur.classList.toggle('red-focus');
344 });
345
346 document.querySelector(DOMstrings.inputBtn).classList.toggle('red');
347
348 },
349
350 getDOMstrings: function() { // exposing DOMstrings to public
351 return DOMstrings;
352 }
353
354 }
355
356 }
```

```
File Edit Selection View Go Run Terminal Help javascriptdemo
EXPLORER index.html U # style.css U JS app.js U X
JS app.js U 356 };
JS app01.js U 357
index.html U 358
# style.css U 359 }());
360
361 // GLOBAL APP CONTROLLER
362 // Controller is the place where we tell the other modules
363 // what to do
364 var controller = (function(budgetCtrl, UIctrl) {
365
366 var setupEventListeners = function() {
367
368 var DOM = UIctrl.getDOMstrings(); // get the DOM strings from UI controller
369
370 document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);
371
372 document.addEventListener('keypress', function(event) {
373
374 // which is for older browsers that don't have this keycode
375 // property
376 if(event.keyCode === 13 || event.which === 13) {
377 ctrlAddItem(); // call the add item function
378 }
379
380 });
381
382 document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);
383
384 // callback function
385 document.querySelector(DOM.inputType).addEventListener('change', UIctrl.changedType);
386
387 };
388
389 var updateBudget = function() {
390
391 //1. calculate the budget
392 budgetCtrl.calculateBudget();
393
394 }
```

```
389 var updateBudget = function() {
390
391     //1. Calculate the budget
392     budgetCtrl.calculateBudget();
393
394     //2. Return the budget
395     var budget = budgetCtrl.getBudget();
396
397     //3. Display the budget on the UI
398     UIctrl.displayBudget(budget);
399
400 };
401
402 var updatePercentages = function() {
403     //1. Calculate percentages
404     budgetCtrl.calculatePercentages();
405
406     //2. Read percentages from the budget controller
407     var percentages = budgetCtrl.getPercentages();
408
409     //3. Update the UI with the new percentages
410     UIctrl.displayPercentages(percentages);
411
412 };
413
414 // function called when someone hits enter key
415 var ctrlAddItem = function() {
416
417     var input, newItem;
418
419     // 1. Get the field input data
420     input = UIctrl.getInput(); // controller calls method then getInput method does something and returns
421
422     if (input.description !== "" && !isNaN(input.value) && input.value > 0) {
423
424         // 2. Add the item to the budget controller
425         newItem = budgetCtrl.addItem(input.type, input.description, input.value);
```

```
426         newItem = budgetCtrl.addItem(input.type, input.description, input.value);
427
428         // 3. Add the item to the UI
429         UIctrl.addItem(newItem, input.type);
430
431         //4. clear the fields
432         UIctrl.clearFields();
433
434         // 5. Calculate and update budget
435         updateBudget();
436
437         // 6. Calculate and update percentages
438         updatePercentages();
439     }
440
441 };
442
443 var ctrlDeleteItem = function(event) { // event there to know what target element is
444     var itemID, splitID, type, ID;
445
446     // parent Node gets the unique ID at the very top
447     itemID = event.target.parentNode.parentNode.parentNode.id; // get parent node of DOM
448     // make things happen ONLY if item id is defined
449     if (itemID) {
450
451         // inc-1
452         splitID = itemID.split('-');
453         type = splitID[0]; //first element is the type
454         ID = parseInt(splitID[1]);
455
456         // 1. Delete item from data structure
457         budgetCtrl.deleteItem(type, ID);
458
459         // 2. Delete the item from the UI
```

```
455 // 1. Delete item from data structure
456 budgetCtrl.deleteItem(type, ID);
457
458 // 2. Delete the item from the UI
459 UIctrl.deleteItem(itemID);
460
461 // 3. Update and show the new budget
462 updateBudget(); // function that updates budget
463
464 // 4. Calculate and update percentages
465 updatePercentages();
466
467 }
468
469 };
470
471
472 return { // event listeners only going to be set up as soon as we call the init function
473   init: function() {
474     // Starts the program HAPPENS AT THE BEGINNING
475     console.log('Application has started');
476     UIctrl.displayMonth();
477     UIctrl.displayBudget(
478       {
479         budget: 0,
480         totalInc: 0,
481         totalExp: 0,
482         percentage: -1
483       }
484     );
485     setupEventListeners();
486   }
487 }
488
489 })(budgetController, UIController);
490
491 controller.init(); // starts the event listeners
```

Use the full code in another pdf attached.