```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="top">
    <div class="budget">
      <div class="budget__title">
        Available Budget in <span class="budget__title--month">%Month%</span>:
      </div>

      <div class="budget__value">+ 2,345.64</div>

      <div class="budget__income clearfix">
        <div class="budget__income--text">Income</div>
        <div class="right">
          <div class="budget__income--value">+ 4,300.00</div>
          <div class="budget__income--percentage">&nbsp;</div>
        </div>
      </div>

      <div class="budget__expenses clearfix">
        <div class="budget__expenses--text">Expenses</div>
        <div class="right clearfix">
          <div class="budget__expenses--value">- 1,954.36</div>
          <div class="budget__expenses--percentage">45%</div>
        </div>
      </div>
    </div>
  </div>


  <div class="bottom">
    <div class="add">
      <div class="add__container">
        <select class="add__type">
                        <option value="inc" selected>+</option>
                        <option value="exp">-</option>
                    </select>
        <input type="text" class="add__description" placeholder="Add description">
        <input type="number" class="add__value" placeholder="Value">
        <button class="add__btn"><i class="ion-ios-checkmark-outline"></i></button>
      </div>
    </div>

    <div class="container clearfix">
      <div class="income">
        <h2 class="income__title">Income</h2>

        <div class="income__list">
        </div>
      </div>


      <div class="expenses">
```

```
        <h2 class="expenses__title">Expenses</h2>

        <div class="expenses__list">
        </div>
      </div>
    </div>
  </div>

  <style>
      /**********************************************
*** GENERAL
**********************************************/

* {
margin: 0;
padding: 0;
box-sizing: border-box;
}

.clearfix::after {
content: "";
display: table;
clear: both;
}

body {
color: #555;
font-family: Open Sans;
font-size: 16px;
position: relative;
height: 100vh;
font-weight: 400;
}

.right {
float: right;
}

.red {
color: #FF5049 !important;
}

.red-focus:focus {
border: 1px solid #FF5049 !important;
}


/**********************************************
*** TOP PART
**********************************************/

.top {
height: 40vh;
background-image: linear-gradient(rgba(0, 0, 0, 0.35), rgba(0, 0, 0, 0.35)),
url("http://www.pamaciba.lv/wp-content/uploads/2014/09/ka-aprekinat-pvn.jpg");
background-size: cover;
background-position: center;
position: relative;
}
```

```
.budget {
position: absolute;
width: 350px;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
color: #fff;
}

.budget__title {
font-size: 18px;
text-align: center;
margin-bottom: 10px;
font-weight: 300;
}

.budget__value {
font-weight: 300;
font-size: 46px;
text-align: center;
margin-bottom: 25px;
letter-spacing: 2px;
}

.budget__income,
.budget__expenses {
padding: 12px;
text-transform: uppercase;
}

.budget__income {
margin-bottom: 10px;
background-color: #28B9B5;
}

.budget__expenses {
background-color: #FF5049;
}

.budget__income--text,
.budget__expenses--text {
float: left;
font-size: 13px;
color: #444;
margin-top: 2px;
}

.budget__income--value,
.budget__expenses--value {
letter-spacing: 1px;
float: left;
}

.budget__income--percentage,
.budget__expenses--percentage {
float: left;
width: 34px;
font-size: 11px;
padding: 3px 0;
margin-left: 10px;
}

.budget__expenses--percentage {
background-color: #ffffff;
background-color: rgba(255, 255, 255, 0.2);
text-align: center;
border-radius: 3px;
}


/**********************************************
*** BOTTOM PART
**********************************************/


/***** FORM *****/

.add {
padding: 14px;
border-bottom: 1px solid #e7e7e7;
background-color: #f7f7f7;
}

.add__container {
margin: 0 auto;
text-align: center;
}

.add__type {
width: 55px;
border: 1px solid #e7e7e7;
height: 44px;
font-size: 18px;
color: inherit;
background-color: #fff;
margin-right: 10px;
font-weight: 300;
transition: border 0.3s;
}

.add__description,
.add__value {
border: 1px solid #e7e7e7;
background-color: #fff;
color: inherit;
font-family: inherit;
font-size: 14px;
padding: 12px 15px;
margin-right: 10px;
border-radius: 5px;
transition: border 0.3s;
}

.add__description {
width: 400px;
}
```

```
.add__value {
width: 100px;
}

.add__btn {
font-size: 35px;
background: none;
border: none;
color: #28B9B5;
cursor: pointer;
display: inline-block;
vertical-align: middle;
line-height: 1.1;
margin-left: 10px;
}

.add__btn:active {
transform: translateY(2px);
}

.add__type:focus,
.add__description:focus,
.add__value:focus {
outline: none;
border: 1px solid #28B9B5;
}

.add__btn:focus {
outline: none;
}


/***** LISTS *****/

.container {
width: 1000px;
margin: 60px auto;
}

.income {
float: left;
width: 475px;
margin-right: 50px;
}

.expenses {
float: left;
width: 475px;
}

h2 {
text-transform: uppercase;
font-size: 18px;
font-weight: 400;
margin-bottom: 15px;
}

.income__title {
color: #28B9B5;
```

```
}

.expenses__title {
color: #FF5049;
}

.item {
padding: 13px;
border-bottom: 1px solid #e7e7e7;
}

.item:first-child {
border-top: 1px solid #e7e7e7;
}

.item:nth-child(even) {
background-color: #f7f7f7;
}

.item__description {
float: left;
}

.item__value {
float: left;
transition: transform 0.3s;
}

.item__percentage {
float: left;
margin-left: 20px;
transition: transform 0.3s;
font-size: 11px;
background-color: #FFDAD9;
padding: 3px;
border-radius: 3px;
width: 32px;
text-align: center;
}

.income .item__value,
.income .item__delete--btn {
color: #28B9B5;
}

.expenses .item__value,
.expenses .item__percentage,
.expenses .item__delete--btn {
color: #FF5049;
}

.item__delete {
float: left;
}

.item__delete--btn {
font-size: 22px;
background: none;
border: none;
```

```
cursor: pointer;
display: inline-block;
vertical-align: middle;
line-height: 1;
display: none;
}

.item__delete--btn:focus {
outline: none;
}

.item__delete--btn:active {
transform: translateY(2px);
}

.item:hover .item__delete--btn {
display: block;
}

.item:hover .item__value {
transform: translateX(-20px);
}

.item:hover .item__percentage {
transform: translateX(-20px);
}

.unpaid {
background-color: #FFDAD9 !important;
cursor: pointer;
color: #FF5049;
}

.unpaid .item__percentage {
box-shadow: 0 2px 6px 0 rgba(0, 0, 0, 0.1);
}

.unpaid:hover .item__description {
font-weight: 900;
}
  </style>

  <script>
      // BUDGET CONTROLLER
var budgetController = (function() {

var Expense = function(id, description, value) {
    this.id = id;
    this.description = description;
    this.value = value;
    this.percentage = -1;
};


Expense.prototype.calcPercentage = function(totalIncome) {
    if (totalIncome > 0) {
        this.percentage = Math.round((this.value / totalIncome) * 100);
    } else {
        this.percentage = -1;
    }
};


Expense.prototype.getPercentage = function() {
    return this.percentage;
};


var Income = function(id, description, value) {
    this.id = id;
    this.description = description;
    this.value = value;
};


var calculateTotal = function(type) {
    var sum = 0;
    data.allItems[type].forEach(function(cur) {
        sum += cur.value;
    });
    data.totals[type] = sum;
};


var data = {
    allItems: {
        exp: [],
        inc: []
    },
    totals: {
        exp: 0,
        inc: 0
    },
    budget: 0,
    percentage: -1
};


return {
    addItem: function(type, des, val) {
        var newItem, ID;

        //[1 2 3 4 5], next ID = 6
        //[1 2 4 6 8], next ID = 9
        // ID = last ID + 1

        // Create new ID
        if (data.allItems[type].length > 0) {
            ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
        } else {
            ID = 0;
        }

        // Create new item based on 'inc' or 'exp' type
        if (type === 'exp') {
            newItem = new Expense(ID, des, val);
        } else if (type === 'inc') {
            newItem = new Income(ID, des, val);
        }

        // Push it into our data structure
        data.allItems[type].push(newItem);

        // Return the new element
        return newItem;
    },


    deleteItem: function(type, id) {
        var ids, index;

        // id = 6
        //data.allItems[type][id];
        // ids = [1 2 4  8]
        //index = 3

        ids = data.allItems[type].map(function(current) {
            return current.id;
        });

        index = ids.indexOf(id);

        if (index !== -1) {
            data.allItems[type].splice(index, 1);
        }

    },


    calculateBudget: function() {

        // calculate total income and expenses
        calculateTotal('exp');
        calculateTotal('inc');

        // Calculate the budget: income - expenses
        data.budget = data.totals.inc - data.totals.exp;

        // calculate the percentage of income that we spent
        if (data.totals.inc > 0) {
            data.percentage = Math.round((data.totals.exp / data.totals.inc) *
100);
        } else {
            data.percentage = -1;
        }

        // Expense = 100 and income 300, spent 33.333% = 100/300 = 0.3333 * 100
    },

    calculatePercentages: function() {

        /*
        a=20
        b=10
        c=40
        income = 100
        a=20/100=20%
        b=10/100=10%
        c=40/100=40%
        */

        data.allItems.exp.forEach(function(cur) {
           cur.calcPercentage(data.totals.inc);
        });
    },


    getPercentages: function() {
        var allPerc = data.allItems.exp.map(function(cur) {
            return cur.getPercentage();
        });
        return allPerc;
    },


    getBudget: function() {
        return {
            budget: data.budget,
            totalInc: data.totals.inc,
            totalExp: data.totals.exp,
            percentage: data.percentage
        };
    },

    testing: function() {
        console.log(data);
    }
};

})();




// UI CONTROLLER
var UIController = (function() {

var DOMstrings = {
    inputType: '.add__type',
    inputDescription: '.add__description',
    inputValue: '.add__value',
    inputBtn: '.add__btn',
    incomeContainer: '.income__list',
    expensesContainer: '.expenses__list',
    budgetLabel: '.budget__value',
    incomeLabel: '.budget__income--value',
    expensesLabel: '.budget__expenses--value',
    percentageLabel: '.budget__expenses--percentage',
    container: '.container',
    expensesPercLabel: '.item__percentage',
    dateLabel: '.budget__title--month'
};


var formatNumber = function(num, type) {
    var numSplit, int, dec, type;
    /*
        + or - before number
        exactly 2 decimal points
        comma separating the thousands
        2310.4567 -> + 2,310.46
        2000 -> + 2,000.00
        */

    num = Math.abs(num);
    num = num.toFixed(2);

    numSplit = num.split('.');

    int = numSplit[0];
    if (int.length > 3) {
        int = int.substr(0, int.length - 3) + ',' + int.substr(int.length - 3,
3); //input 23510, output 23,510
    }

    dec = numSplit[1];

    return (type === 'exp' ? '-' : '+') + ' ' + int + '.' + dec;

};


var nodeListForEach = function(list, callback) {
    for (var i = 0; i < list.length; i++) {
        callback(list[i], i);
    }
};


return {
    getInput: function() {
        return {
            type: document.querySelector(DOMstrings.inputType).value, // Will be
either inc or exp
            description: document.querySelector(DOMstrings.inputDescription).value,
            value: parseFloat(document.querySelector(DOMstrings.inputValue).value)
        };
    },


    addListItem: function(obj, type) {
        var html, newHtml, element;
        // Create HTML string with placeholder text

        if (type === 'inc') {
            element = DOMstrings.incomeContainer;

            html = '<div class="item clearfix" id="inc-%id%"> <div
class="item__description">%description%</div><div class="right clearfix"><div
class="item__value">%value%</div><div class="item__delete"><button
class="item__delete--btn"><i
class="ion-ios-close-outline"></i></button></div></div></div>';
        } else if (type === 'exp') {
            element = DOMstrings.expensesContainer;
            html = '<div class="item clearfix" id="exp-%id%"><div 
class="item__description">%description%</div><div class="right clearfix"><div 
class="item__value">%value%</div><div class="item__percentage">21%</div><div 
class="item__delete"><button class="item__delete--btn"><i class="ion-ios-close-
outline"></i></button></div></div></div>';
        }

        // Replace the placeholder text with some actual data
        newHtml = html.replace('%id%', obj.id);
        newHtml = newHtml.replace('%description%', obj.description);
        newHtml = newHtml.replace('%value%', formatNumber(obj.value, type));

        // Insert the HTML into the DOM
        document.querySelector(element).insertAdjacentHTML('beforeend', newHtml);
    },


    deleteListItem: function(selectorID) {

        var el = document.getElementById(selectorID);
        el.parentNode.removeChild(el);

    },


    clearFields: function() {
        var fields, fieldsArr;

        fields = document.querySelectorAll(DOMstrings.inputDescription + ', ' + 
DOMstrings.inputValue);

        fieldsArr = Array.prototype.slice.call(fields);

        fieldsArr.forEach(function(current, index, array) {
            current.value = "";
        });

        fieldsArr[0].focus();
    },


    displayBudget: function(obj) {
        var type;
        obj.budget > 0 ? type = 'inc' : type = 'exp';

        document.querySelector(DOMstrings.budgetLabel).textContent = 
formatNumber(obj.budget, type);
        document.querySelector(DOMstrings.incomeLabel).textContent = 
formatNumber(obj.totalInc, 'inc');
        document.querySelector(DOMstrings.expensesLabel).textContent = 
formatNumber(obj.totalExp, 'exp');

        if (obj.percentage > 0) {
            document.querySelector(DOMstrings.percentageLabel).textContent = 
obj.percentage + '%';
        } else {
            document.querySelector(DOMstrings.percentageLabel).textContent = '---';
        }
    },


    displayPercentages: function(percentages) {

        var fields = document.querySelectorAll(DOMstrings.expensesPercLabel);

        nodeListForEach(fields, function(current, index) {

            if (percentages[index] > 0) {
                current.textContent = percentages[index] + '%';
            } else {
                current.textContent = '---';
            }
        });

    },


    displayMonth: function() {
        var now, months, month, year;

        now = new Date();
        //var christmas = new Date(2016, 12, 25);

        months = ['January', 'February', 'March', 'April', 'May', 'June', 'July',
'August', 'September', 'October', 'November', 'December'];
        month = now.getMonth();

        year = now.getFullYear();
        document.querySelector(DOMstrings.dateLabel).textContent = months[month] +
' ' + year;
    },


    changedType: function() {

        var fields = document.querySelectorAll(
            DOMstrings.inputType + ',' +
            DOMstrings.inputDescription + ',' +
            DOMstrings.inputValue);

        nodeListForEach(fields, function(cur) {
           cur.classList.toggle('red-focus');
        });

        document.querySelector(DOMstrings.inputBtn).classList.toggle('red');

    },


    getDOMstrings: function() {
        return DOMstrings;
    }
};

})();
```

```
// GLOBAL APP CONTROLLER
var controller = (function(budgetCtrl, UICtrl) {

var setupEventListeners = function() {
    var DOM = UICtrl.getDOMstrings();

    document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

    document.addEventListener('keypress', function(event) {
        if (event.keyCode === 13 || event.which === 13) {
            ctrlAddItem();
        }
    });

    document.querySelector(DOM.container).addEventListener('click',
ctrlDeleteItem);

    document.querySelector(DOM.inputType).addEventListener('change',
UICtrl.changedType);
};


var updateBudget = function() {

    // 1. Calculate the budget
    budgetCtrl.calculateBudget();

    // 2. Return the budget
    var budget = budgetCtrl.getBudget();

    // 3. Display the budget on the UI
    UICtrl.displayBudget(budget);
};


var updatePercentages = function() {

    // 1. Calculate percentages
    budgetCtrl.calculatePercentages();

    // 2. Read percentages from the budget controller
    var percentages = budgetCtrl.getPercentages();

    // 3. Update the UI with the new percentages
    UICtrl.displayPercentages(percentages);
};


var ctrlAddItem = function() {
    var input, newItem;

    // 1. Get the field input data
    input = UICtrl.getInput();

    if (input.description !== "" && !isNaN(input.value) && input.value > 0) {
        // 2. Add the item to the budget controller
        newItem = budgetCtrl.addItem(input.type, input.description, input.value);
        // 3. Add the item to the UI
        UICtrl.addListItem(newItem, input.type);

        // 4. Clear the fields
        UICtrl.clearFields();

        // 5. Calculate and update budget
        updateBudget();

        // 6. Calculate and update percentages
        updatePercentages();
    }
};


var ctrlDeleteItem = function(event) {
    var itemID, splitID, type, ID;

    itemID = event.target.parentNode.parentNode.parentNode.parentNode.id;

    if (itemID) {

        //inc-1
        splitID = itemID.split('-');
        type = splitID[0];
        ID = parseInt(splitID[1]);

        // 1. delete the item from the data structure
        budgetCtrl.deleteItem(type, ID);

        // 2. Delete the item from the UI
        UICtrl.deleteListItem(itemID);

        // 3. Update and show the new budget
        updateBudget();

        // 4. Calculate and update percentages
        updatePercentages();
    }
};


return {
    init: function() {
        console.log('Application has started.');
        UICtrl.displayMonth();
        UICtrl.displayBudget({
            budget: 0,
            totalInc: 0,
            totalExp: 0,
            percentage: -1
        });
        setupEventListeners();
    }
};

})(budgetController, UIController);
```

```
controller.init();
  </script>

</body>
</html>
```