//BUDGET CONTROLLER // keeps track of all the expenses and incomes and of the budget itself and the percentages

var budgetController = (function() {

var Expense = function(id, description, value) {

this.id = id;

this.description = description;

this.value = value;

this.percentage = -1;

};

//method of calculating percentages - this is calculating

Expense.prototype.calcPercentage = function(totalIncome) {

if(totalIncome > 0) {

this.percentage = Math.round ((this.value / totalIncome) \* 100);

} else {

this.percentage = -1;

}

};

/\*method of getting percentages - the simplest one, only to retrieving the % from the object - this is returning the percentages\*/

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 = { //in acest OB se stocheaza toate datele

allItems: { // toate cheltuielle si veniturile ce vor veni

exp: [], //cheltuielile ce vor veni, array gol

inc:[] //venituri ce vor veni, array gol

},

totals: { //totalurile de cheltuieli si venituri

exp: 0, //totaluri cheltuieli

inc: 0 //totaluri venituri

},

budget: 0,

percentage: -1

};

return {

addItem: function(type, des, val) {

var newItem, ID;

/\*id = [1, 2, 5, 6, 8] next id is 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 "exp" si "inc" type

if(type === "exp") {

newItem = new Expense(ID, des, val);

} else if (type ==="inc") {

newItem = new Income(ID, des, val);

}

//push it onto our data structure

data.allItems[type].push(newItem);

//return the new element

return newItem;

},

deleteItem: function(type, id) {

var ids, index;

ids = data.allItems[type].map(function(current) {

return current.id;

});

index = ids.indexOf(id);

if (index !== -1) {

data.allItems[type].splice(index, 1);

// index = nr de sters, 1 = stergem doar o pozitie

}

},

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;

}

},

calculatePercentages: function() {

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;

// 2 decimals

num = Math.abs(num);

num = num.toFixed(2);

// .after 2 decimals

numSplit = num.split(".");

//, after thousands

//for ex 2310, 4567 - 2,310.46

int = numSplit[0];

if (int.length > 3) {

int = int.substr(0, int.length - 3) + "," + int.substr(int.length - 3, 3);

}

dec = numSplit[1];

//+ or - befor num

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, //it can 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 strings 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);

/\* a list is an array but without the nice methods of the arrays. solution - converting the list to an array. and to use the slice array method - returns copy of the array. usually this method returns another array but there is a trick to pass a list on it and then returns an array. but fields.slice(); will not work. will use the slice meth using the call method then passing the fields method into it to become the this variable then is gonna work just fine. and where is stored the slice meth that we can use it? in the array prototype. this Array is the function constructor for all the arrays and all the methods the array inherits from the array function constructor are in arrays prototype property. the slice meth is also there. array.prototype.slice is a function that we can call. si we add .call and as argument, the this variable of the fields. il stocam in variabila fieldsArr si aceasta va deveni si ea un array \*/

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

/\*aceasta metoda .foreach loops over all of the elements of the fieldsArray and then sets the value for all of them back to the empty string. and we have access to the current element through this callback function who has access to these 3 arguments: current, index and to the original array which in this case is the fields array \*/

fieldsArr.forEach(function(current, index, array) {

current.value = "";

})

/\*pt ca cursorul sa revina pe partea de description folosim metoda focus() a arraylui pus la description, adica pozitia 0 a sa \*/

fieldsArr[0].focus();

},

displayBudget: function(object) {

var type;

object.budget > 0 ? type = "inc" : type = "exp";

document.querySelector(DOMstrings.budgetLabel).textContent = formatNumber(object.budget, type);

document.querySelector(DOMstrings.incomeLabel).textContent = formatNumber(object.totalInc, "inc");

document.querySelector(DOMstrings.expensesLabel).textContent = formatNumber (object.totalExp, "exp");

if(object.percentage > 0) {

document.querySelector(DOMstrings.percentageLabel).textContent = object.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(2019, 11, 25)

//we have methods to retrieve the current month, date, minute etc

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 setupEventListener = function() {

var DOM = UICtrl.getDOMstrings();

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

// e un evlist general, pt orice keypress pe pagina(nu pt o clasa anume) si se scrie astfel:

document.addEventListener("keypress", function(event) {

if (event.keyCode === 1 || 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 = budgetController.addItem(input.type, input.description, input.value);

//3. Add the item to the UI

UIController.addListItem(newItem, input.type);

// 4. Clear the fields

UIController.clearFields();

//5. update budget

updateBudget();

//6. calculate and update the 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

UIController.deleteListItem(itemID);

//3. update and show the new budget

updateBudget();

//4. calculate and update the percentages

updatePercentages();

}

};

return {

init: function() {

console.log("application has started.");

UICtrl.displayMonth();

UICtrl.displayBudget({

budget: 0,

totalInc: 0,

totalExp: 0,

percentage: -1

});

setupEventListener();

}

};

})(budgetController, UIController);

controller.init();

function getWeather (woeid) {

fetch('https://crossorigin.me/https://www.metaweather.com/api/location/${`woeid`}/')

.then(result => {

console.log(result);

return result.json();

})

.then(data => {

//console.log(data);

const today = data.consolidated\_weather[0];

console.log(`Temperature in ${data.title}) stay between ${today.min\_temp} and ${today.max\_temp}.`)

})

.catch(error => console.log(error));

}

getWeather(2487956);

getWeather(784523111111);