EECS 428 / ECE 578
DATA VISUALIZATION
Spring 2017

ASSIGNMENT 1
Due Date: Monday, *13/03/2017 23:59*
(12 Points)

Assignment Submission: *Turn in your assignment by the due date through LMS**.** Prepare and upload **one zip file** that you will name as **<your first name>_<your last name>_ assignment1**. See individual questions for what you should return.*

You can and encouraged to discuss HTML, CSS, SVG, JavaScript and D3 with each other. ***However, all work in questions (implementation) must be your own; you must neither copy from nor provide assistance to anybody else (including online resources). If you need guidance for any question, talk to the instructor or TAs.***

Budget Tracker

In this assignment, you will implement a simple budget tracker to keep track of the budget and expenditure of a user over a period of time. You may assume that the user has a fixed set of possible expenses that fall under three categories, i.e., sports, luxury and food items. A summary of these expenses is given in Figure 1.1 below.

Sports	**Luxury**	**Food**
Tracksuit - $300	Sunglasses - $150	Apple - $2
Shoes - $100	Watch - $1000	Cucumber - $1
Racket - $200	Necklace - $500	Chicken - $10
Tennis Ball - $20	Coffee - $5	Rice - $15

Figure 1.1 – A list of expenses

You will use HTML, CSS and SVG to design the interface of budget tracker and JavaScript will enable you to implement its functionality. In addition, you will also use D3 to perform some basic visualization tasks.

Part A: Design and Functionality

When your application starts, the initial budget and total expenses of the user should be equal to 0. The user should then be able to enter his initial budget in a text box (see Figure 1.2 below), and when he clicks the **set** button, his **initial budget** should be updated to the value he entered. Also note that the **remaining budget** is calculated as the difference between the initial budget and **total expenses** (initially 0) of the user, and this value should

be updated upon setting the initial budget as well. When the user clicks the **reset** button, the initial budget, total expenses and the remaining budget should all be set to 0.

Figure 1.2 – Setting the initial budget

Following this, your budget tracker should give the user a means to enter and update his total expenses. The design of this mechanism should resemble the one shown in Figure 1.3 below:

Figure 1.3 – Expense manager for updating total expenses

You may use buttons or SVG elements to represent each item in Figure 1.3. Make sure that items in different categories have different colors, for instance in Figure 1.3, sports items are represented with a green color, luxury items are represented with an orange color, etc. The workflow of this expense manager will be as follows:

First the user utilizes the selector (plus or minus buttons) to choose the number of items of a particular expense he wishes to enter (number of items can be negative as well). This number should be reflected in the textbox as shown in Figure 1.3, and its default value should be 1. After selecting the number of items, the user should then click on the actual expense (any item in the expense manager), and following that, **total expenses** should be incremented by (number of items * price of expense) and the remaining budget is updated accordingly. As an example, the user may choose 2 items, and then click on Tracksuit (which has a price of $300). Total expenses should therefore be incremented by $600 following this operation, and the **remaining budget** should be updated accordingly as well.

The user may, however, enter an expense that falls under one of the three categories but is not in the list of the predefined expenses. Such expenses are labelled as **other**. For instance, if the user wants to enter an expense for Oranges, since they are not in the list, he should first select the number of oranges using the plus or minus buttons, enter their price in the Price-Tag textbox (as shown in Figure 1.3) and click on **other** under the **food** category. Following this, the total expenses and remaining budget should be updated by the appropriate amount.

Note that when the remaining budget is less than 0 (negative budget), you should change the background color of the label representing this value to red.

Finally, you are going to design a color palette that the user may use to adjust the color of the items in each category. This palette will look like similar to the one shown in Figure 1.4 below (feel free to use different colors as long as you have a good selection):

Figure 1.4 – Rainbow Color Palette

If the user clicks on a color and then click a category after that, the color of the items in that category should be changed to the one that the user clicked. For instance, items in the sports category are currently represented by the green color in Figure 1.3. If the user clicks on the blue color, then click on the sports label/button, the color of all items under the sports category should change to blue. **You can use D3 for this task.**

The overall look of your budget tracker will be like the one shown in Figure 1.5 below:

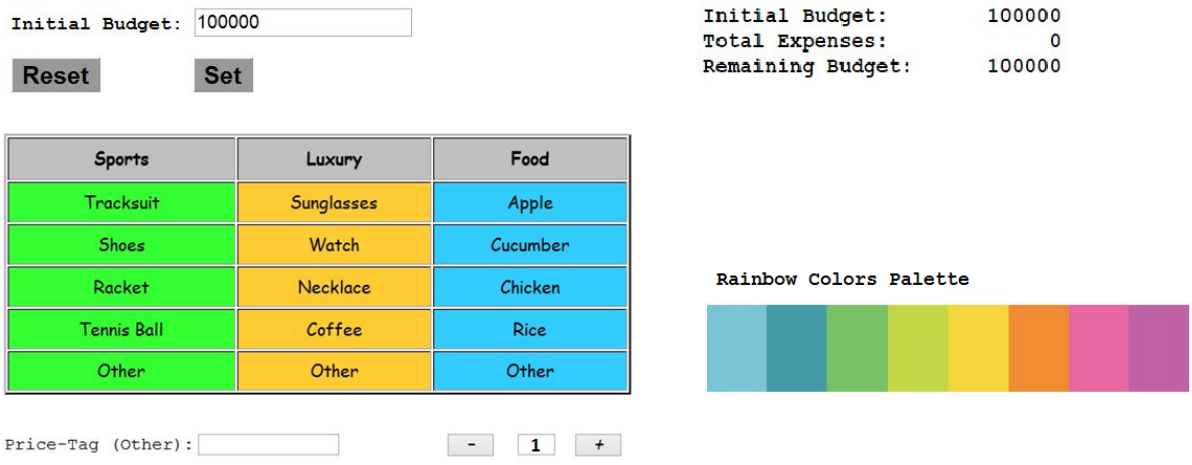

Figure 1.5 – Budget tracking application

Part B: Visualization

For this section, you will plot a bar chart below or next to the budget tracker so as to visualize the proportion of expenses spent by the user in each category. Your bar chart should look like the one shown in Figure 1.6 below:

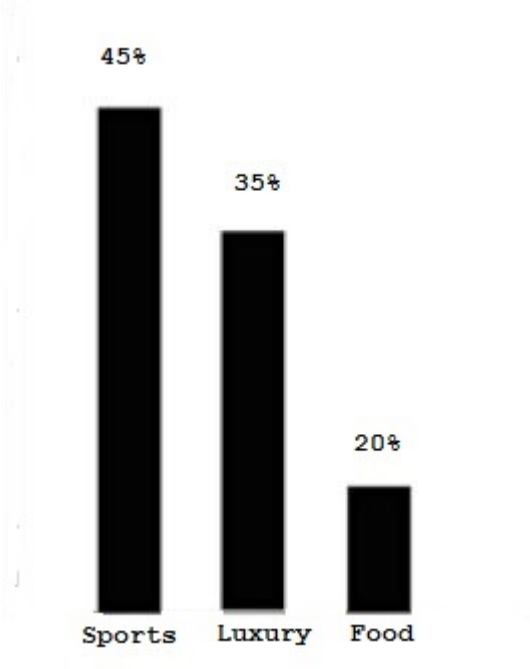

Figure 1.6 – Bar Chart showing proportion of expenses spent in each category

As the user adds his expenses by using the expense manager, the bar chart (3-bars only) should be dynamically updated to reflect any changes in the proportion of expenses spent in each category. Although this can also be implemented without using D3, **you are required to use D3 for this task in this assignment.**

Notes:

- You are not obliged to produce the exact images shown in the illustrations above. In fact you are encouraged to use your creativity to create your own designs as long as it supports the intended functionality and the visually it is at least as appealing as the examples ☺
- You can put all of your code in a single HTML file, but it is highly discouraged. Try to separate your code files: HTML, JavaScript and CSS.

Please return in the zip file:

- All your HTML, JavaScript and CSS files (you can have sub-directories if you prefer). Your main HTML file should be named as **index.html**. When we open index.html in our web browser (Chrome will be used for grading), your budget tracker and bar chart should show up in the main page.