

Francisco Samayoa

3165936

Mobile & Web Cross-Platform Development

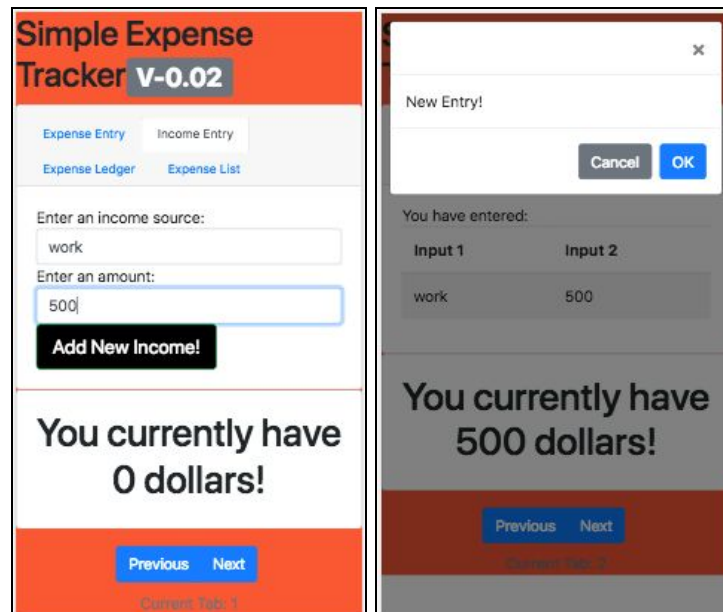
November 5, 2018

Development Assignment

Q1 (10pts) - Using Bootstrap-Vue, design a small application that takes in input from the user and presents it in a table. Use the given starter code (see the Files section, Additional Files folder, for the file from class called "dialog1.html") or build your own:

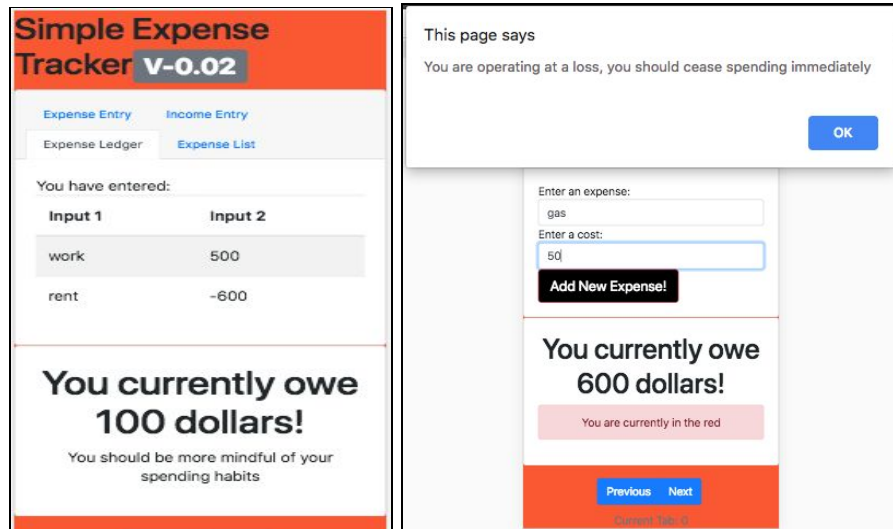
<https://github.com/avongarde/Mobile-Web-Cross-Platform-Development/blob/master/Development%20Assignment/dialog1.html>

- Add tabs for also adding income rather than only expenses
 - Keep track of income as well



Income Entry in action

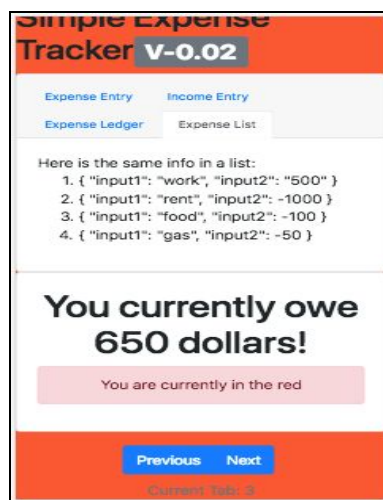
- Calculate the total net income or loss and display a message about how much is being spent or saved
 - If spending is a little more than savings give a gentle warning
 - If spending is a lot more than savings give a strong warning



Gentle Warning (via Expense Ledger)

Strong Warning (via Expense Entry)

- Screenshot using the Galaxy S5 layout in the inspector window:
 - Format or adapt the style of this code using the Bootstrap-Vue tools



Expense List in action

Q2 (5pts) - Explore how you could recreate this in Nativescript and write a short summary of what you would use in your application, based on the Nativescript Playground tutorials and tools. Show a potential mockup of the corresponding Nativescript template.

First I would have to figure out which Vue components correspond to which Nativescript component.

Since Vue follows the same logic as Javascript it shouldn't be too difficult and I don't have to learn any new languages or workflows. I decided to reference a Nativescript template called [Local Data Storage](#) because it looked the most applicable for this situation. It uses scroll view instead of the tab view from the example. I decided to experiment with this view and what I could accomplish. Unfortunately I could not add tabs, but I substituted them with buttons instead. Design wise, I personally enjoy this view more than the tab view because it looks more condensed and aesthetically pleasing.

The image displays three sequential screenshots of a mobile application titled "Simple Expense Tracker v0.02" running in the Nativescript Playground environment. The status bar at the top of each screen shows "Playground", signal strength, Wi-Fi, the time (7:03 PM, 7:04 PM, and 7:20 PM respectively), and a 22% battery level.

First Screenshot (7:03 PM): This screen shows the initial state of the app. It features four blue buttons in a 2x2 grid: "Expense Entry", "Income Entry", "Expense Ledger", and "Expense List". Below these buttons, there are two input sections. The first, "Enter an expense", has a text input field labeled "Name...". The second, "Enter a cost", has a text input field labeled "Cost...". A blue "Submit" button is positioned below the cost input. At the bottom of the screen is a wide blue button labeled "Reset Inputs".

Second Screenshot (7:04 PM): This screen is identical to the first, but the "Enter an income source" section is visible instead of the expense section. It features a text input field labeled "Name..." and a blue "Submit" button below it. The "Enter an amount" section has a text input field labeled "Cost..." and a blue "Submit" button below it. The "Reset Inputs" button remains at the bottom.

Third Screenshot (7:20 PM): This screen shows the result of the user's input. The "Expense Ledger" button has been selected, displaying a list of expenses in a scrollable view. The list contains four entries: "Work" with a value of "500", "Rent" with "-1000", "Food" with "-100", and "Gas" with "-50". Below the list, a message states "You currently owe 650 dollars!". A blue "Reset Inputs" button is located at the bottom of the screen.