

Documentation for Budget App

Introduction

This is a simple budgeting app used to calculate user balance, where users can add and delete their expenses.

The address of the app is <http://budget-app-renee-li.s3-website-ap-southeast-2.amazonaws.com>

Key Features

- This is a serverless website. It is developed by using Angular 6, AWS S3, Lambda, API gateway and DynamoDB.
- In this app, user can do the following thing:
 - Users can view their previous expenses.
 - Users can add new expenses including item name and cost.
 - Users can delete an expense.
 - When users click the edit button, the corresponding expense object data is fetched.
 - Balance will be calculated automatically.

Issues & solution

Due to the app being developed within 4 hours, there are some issues present.

Hard coding in the value of Income

Issues

Currently, the value of income is hard coded as \$2000. User is not able to edit it.

Solution

- create another table called IncomeTable in DynamoDB to store the income data
- add edit, delete and add functions for the incomeTable in Lambda
- add edit, delete and add functions in the app

No authentication in the app

Issues

There is no user login feature in the app. It is open to the public, so everyone is able to modify the data in the database

Solution

- create a new table in DynamoDB to store users' data including login name and password
- Add a login page and a JWT authentication service in the app to user's credentials to the API and checks the response for a JWT token.
- Add a user model to defines the properties of user

Edit function

Issues

There is an edit function, which enables users edit existing expense. Currently, when users click the edit function, users are directed to the edit-cost.component.html, the corresponding exID are passed into the edit-cost.component.html, so user does not need to enter the exID.

However, there is a bug in this function, which is *"Method PUT is not allowed by Access-Control-Allow-Methods in preflight response."*

Solution

- debug the edit function in AWS Lambda and the setting in API gateway.
- debug the updateExpense function in the expense.service.ts.

Validation

Issues

There are no validations on the expense input form. If the user entered invalid data in the cost section, it can break the balance calculation function.

Solution

- add validators into the form to ensure user can only enter the valid number
- add the catch into the balance calculation function to handle unexpected outcome

Expense ID is not auto generated

Issues

When adding a new expense, users need to give a unique id to the expense.

Solution

- add a function to fetch the biggest number of the exID.
- Create a new variable called theNewExID. The value of theNewExID is the next number of the number fetching from step 1.
- Pass theNewExID by localStorage to the add expense form.

Would Features

If there is more time, I will add the followings features:

- Edit, add, delete income sources
- Download the data as an csv file
- Upload data from a file
- Sorting function to sort the expense by ID, item or cost
- Search function to search an expense
- Restyle the app. Make it responsive.
- Make the app more mobile-friendlier by using PWA, so users can download the app onto their mobile's home screen.
- Add different taps to display weekly budget and monthly budget.
- Make a chart to display the difference between different time periods.