

PERSONAL BUDGET PLANNER

Author: Kumiren Naidoo

[READ ME](#)

Table of contents

Description.....	2
Software required:	2
Hardware requirements:	2,3
Running the application in visual studio 2019.....	3 4
Run application file.....	4,5
Application Use.....	6,7,8,9,10,11
Changes made based on feedback.....	12
Support.....	13

Description

This application is a personal budget planner. It allows users to enter an income and month for this income. This allows for multiple budgets to be made for different months, during run time. There are 5 screens available to be accessed. The first is to enter an income and month as discussed before. There are 3 screens for expense input. These are generic expenses, vehicle expense and property expense. Generic expenses consist of tax deductions, Groceries, Water and lights, Travel costs, Cell phone/telephone and Other expenses. In property expense, you may choose rent which consists of Monthly rental amount or buying which consists of Purchase price of property, Total deposit, Interest rate and Number of months to repay. A vehicle expense consists of make, model, purchase price, total deposit, interest rate and insurance premium. The application will perform calculations and display a budget report for the user with the total money left over after deducting all expenses. The total money left is updated after every expense is made. A message is displayed if a users Home loan is likely to be approved, if they selected to buy a property. A message is also displayed when the total expenses exceed 75% of the users income. This message will be displayed after every Expense made, where 75% of the income is exceeded. This is done to continually warn a user that the expenses exceed 75% of the income because they may make an expense that is excessive, however forget then make another expense further exceeding the 75% mark, thus the application continually alerts the user so they do not forget. The 5th screen offers the user an option to view a report of all budgets calculated during run time. This application persists no data between runs.

Software required:

- Visual Studio 2019

Hardware requirements:

You will need the following hardware requirements to run Visual studio 2019

- 1.8 GHz or faster processor. Quad-core or better recommended.
- 2 GB of RAM; 8 GB of RAM recommended
- Hard disk Space: Minimum of 800MB up to 210 GB of available space.

To successfully run this application, you will require Visual Studio 2019 on your device with the following Workloads attached: .NET desktop development & Universal Windows Platform development.

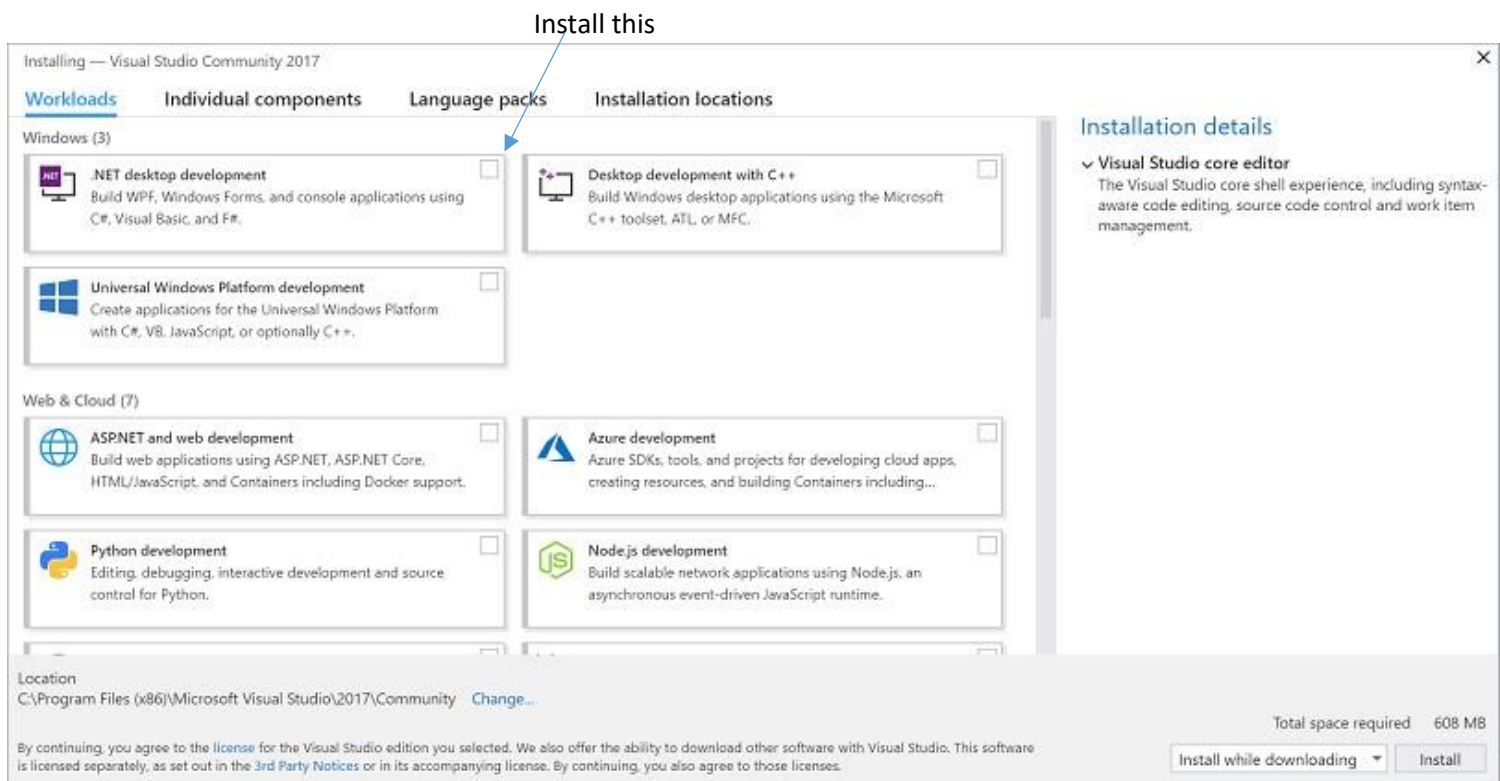
Installation:

Link for the installation for Visual Studio 2019 :

<https://visualstudio.microsoft.com/downloads/> Link for the installation tutorial for

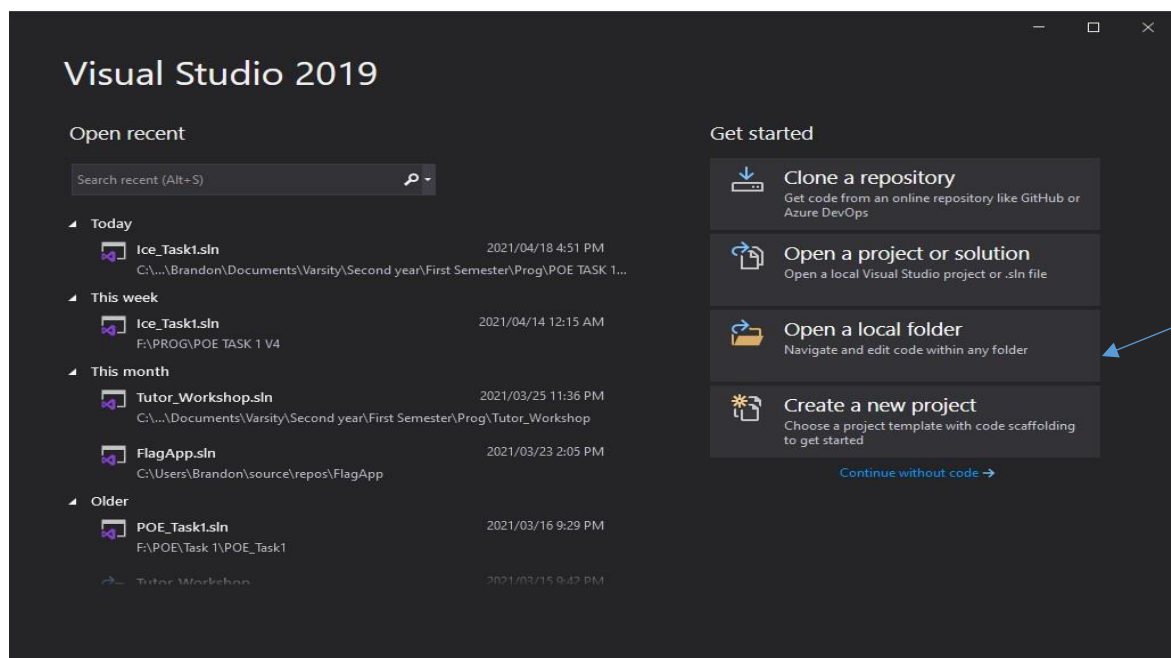
Visual Studio 2019: <https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2019>

Under workloads when installing be sure to select “.Net desktop development”



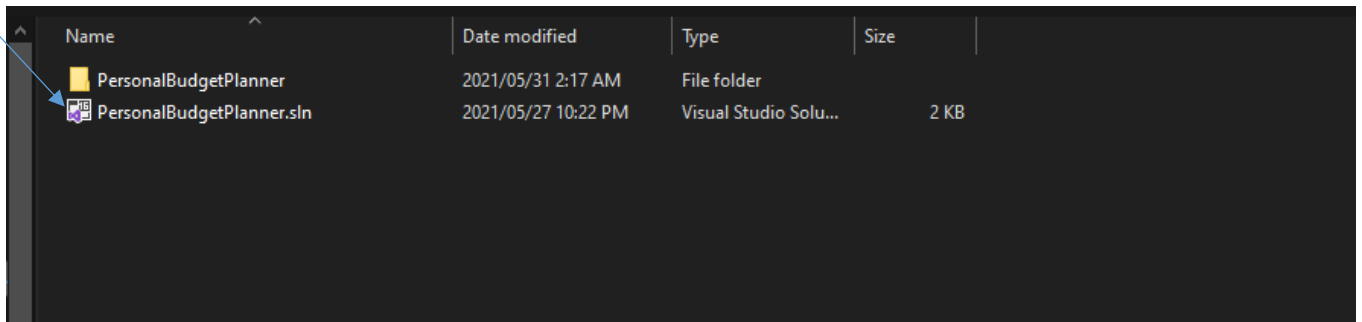
Running the application in visual studio 2019

1. Open Visual Studio 2019

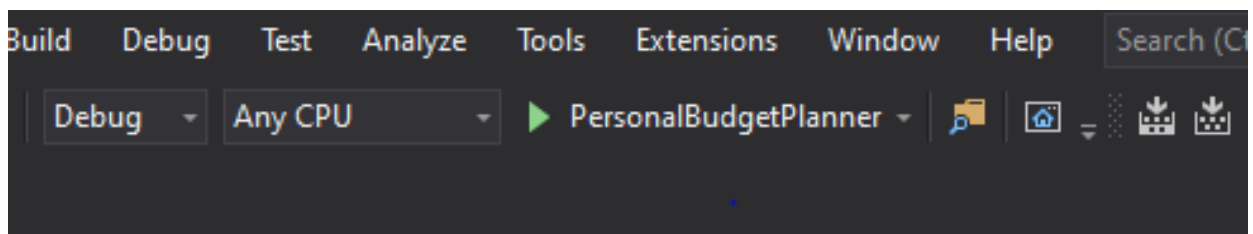


2. Click “Open a project or solution”

3. Unzip if necessary and open the “Budget app” folder then select the “PersonalBudgetPlanner.sln” file

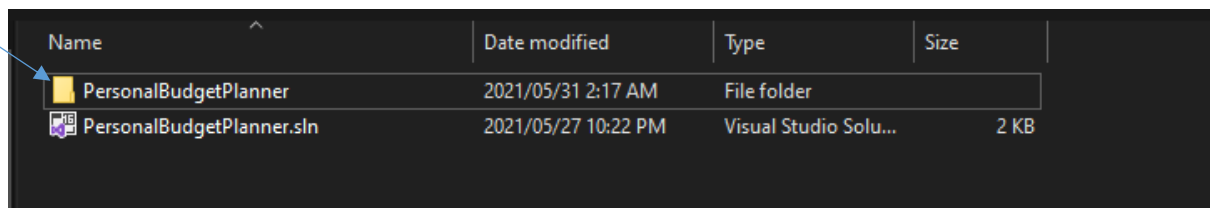


4. When it opens in Visual Studio, click “PersonalBudgetPlanner” with the green play button next to it

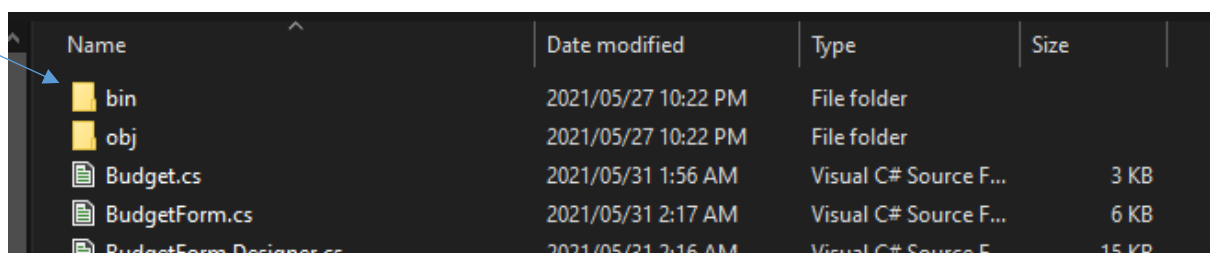


Run application file

1. Unzip if necessary and open the “PersonalBudgetPlanner” folder
2. Open the “PersonalBudgetPlanner” folder



3. Open the “bin” folder



4. Open the “Debug” folder

Name	Date modified	Type	Size
Debug	2021/05/27 10:22 PM	File folder	

5. Open the “netcoreapp3.1” folder

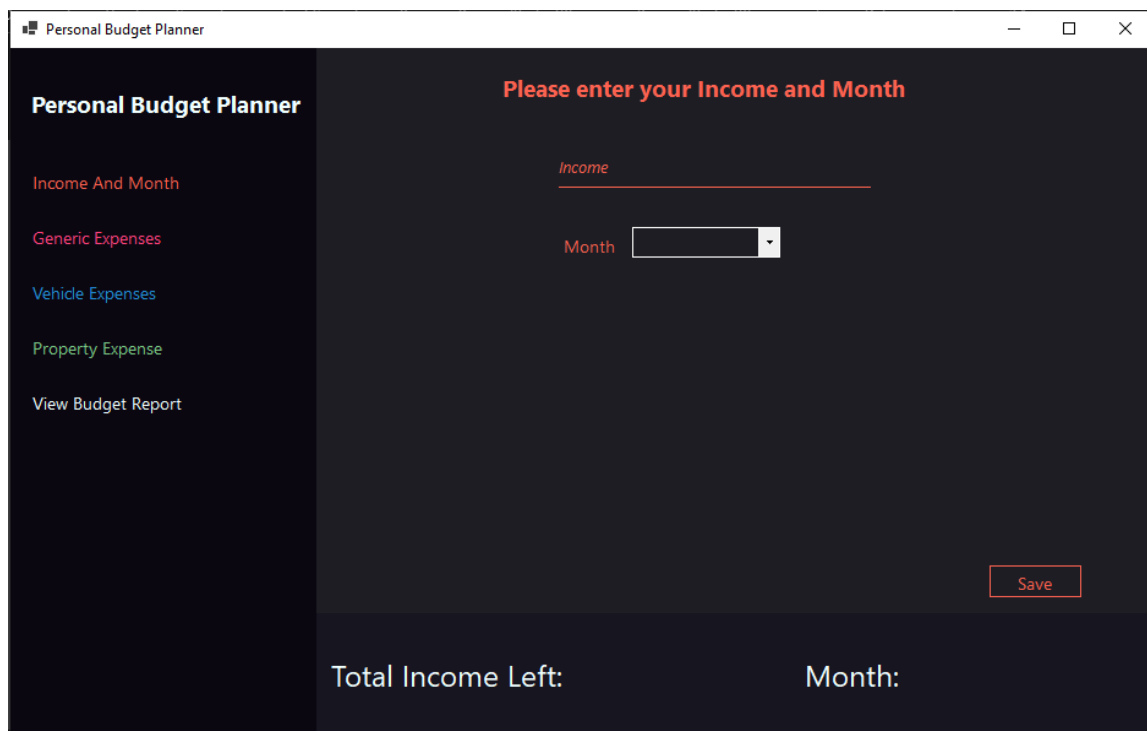
Name	Date modified	Type	Size
netcoreapp3.1	2021/05/27 11:56 PM	File folder	

6. Open the “PersonalBudgetPlanner” application file

Name	Date modified	Type	Size
PersonalBudgetPlanner.deps	2021/05/28 11:19 PM	JSON File	1 KB
PersonalBudgetPlanner.dll	2021/05/31 2:18 AM	Application exten...	51 KB
PersonalBudgetPlanner	2021/05/31 2:18 AM	Application	171 KB
PersonalBudgetPlanner.pdb	2021/05/31 2:18 AM	Program Debug D...	29 KB
PersonalBudgetPlanner.runtimeconfig.dev	2021/05/27 11:56 PM	JSON File	1 KB
PersonalBudgetPlanner.runtimeconfig	2021/05/27 11:56 PM	JSON File	1 KB

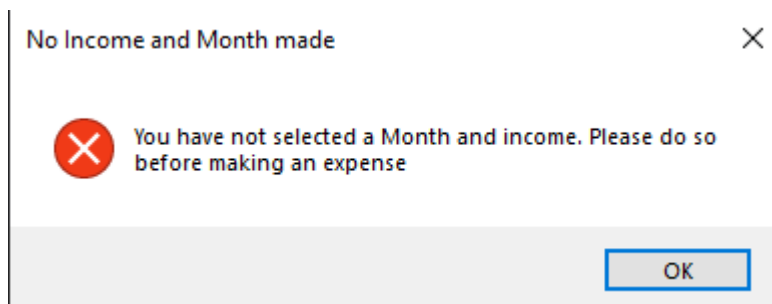
Application Use

When you run the application, this screen appears



The screenshot shows a window titled "Personal Budget Planner". On the left is a dark sidebar with the title "Personal Budget Planner" and five menu items: "Income And Month" (highlighted in red), "Generic Expenses" (red), "Vehicle Expenses" (blue), "Property Expense" (green), and "View Budget Report" (white). The main area has a dark background with the red text "Please enter your Income and Month". Below this, there is a red label "Income" followed by a text input field, and a red label "Month" followed by a dropdown menu. A red "Save" button is in the bottom right. At the bottom, there are two labels: "Total Income Left:" and "Month:".

You will be required to enter a gross monthly income and a month for which the budget will be made, before accessing any other features. The following message will be displayed if you do not do this.



An income and month input should look similar to this

The screenshot shows a web application titled "Personal Budget Planner". On the left is a dark sidebar with the title "Personal Budget Planner" and five menu items: "Income And Month" (highlighted in red), "Generic Expenses" (pink), "Vehicle Expenses" (blue), "Property Expense" (green), and "View Budget Report" (white). The main content area has a dark background with the heading "Please enter your Income and Month" in red. Below this heading are two input fields: "Income" with the value "10000" and "Month" with a dropdown menu showing "April". A red "Save" button is located in the bottom right of the main area. At the bottom of the application, there is a dark footer bar with the text "Total Income Left:" and "Month:".

When you are satisfied with your values click “Save” and your income and month will be saved and the screen will look like the following, after clicking “Save”

This screenshot shows the same application after the "Save" button was clicked. The sidebar remains the same. The main content area still has the heading "Please enter your Income and Month". The "Income" field now has the label "Income" above it and contains the value "10000". The "Month" dropdown menu now shows "April" as the selected option. The red "Save" button is still present. The footer bar now displays the saved values: "Total Income Left: 10000" and "Month: April".

If you made an error in your input, you may input the proper values and click “Save” again to overwrite the budget and save these proper values

Input proper values.
Then click “Save”

The screenshot shows a web application titled "Personal Budget Planner". On the left is a dark sidebar with the title and five menu items: "Income And Month", "Generic Expenses", "Vehicle Expenses", "Property Expense", and "View Budget Report". The main content area has a dark background with the heading "Please enter your Income and Month" in red. Below this heading, there is a text input field containing "100000" and a dropdown menu for "Month" with "September" selected. A red "Save" button is in the bottom right. At the bottom of the page, a status bar shows "Total Income Left: 10000" and "Month: April".

This screenshot shows the same application after a save action. The input field now contains "Income" and the "Month" dropdown is empty. The status bar at the bottom now displays "Total Income Left: 100000" and "Month: September". A blue arrow points from the text "Proper values are saved and old ones deleted" to the updated status bar.

Proper values
are saved
and old ones
deleted

Please note. If an input is incorrect the line under the input will show red. Hover over this line for feedback on what the error is. This is true for all expenses aswell

Personal Budget Planner

Personal Budget Planner

Income And Month

Generic Expenses

Vehicle Expenses

Property Expense

View Budget Report

Please enter your Income and Month

Please enter a value!

Month

Save

Total Income Left: Month:

You can now make any type of expense. Here is an example of a Generic expense input and once it is saved

Personal Budget Planner

Generic Expenses

1000

400

2000

150

790

620

Save

Total Income Left: 10000 Month: April

Personal Budget Planner

Income And Month

Generic Expenses

Vehicle Expenses

Property Expense

View Budget Report

Generic Expenses

Tax Deductions

Groceries

Water & Lights

Travel((inc. Petrol)

Cellphone & Telephone

Other

GenericExpenses:4960

Water&Lights: 2000

Tax Deductions: 1000

CellTell: 790

Other: 620

Groceries: 400

Travel: 150

Save

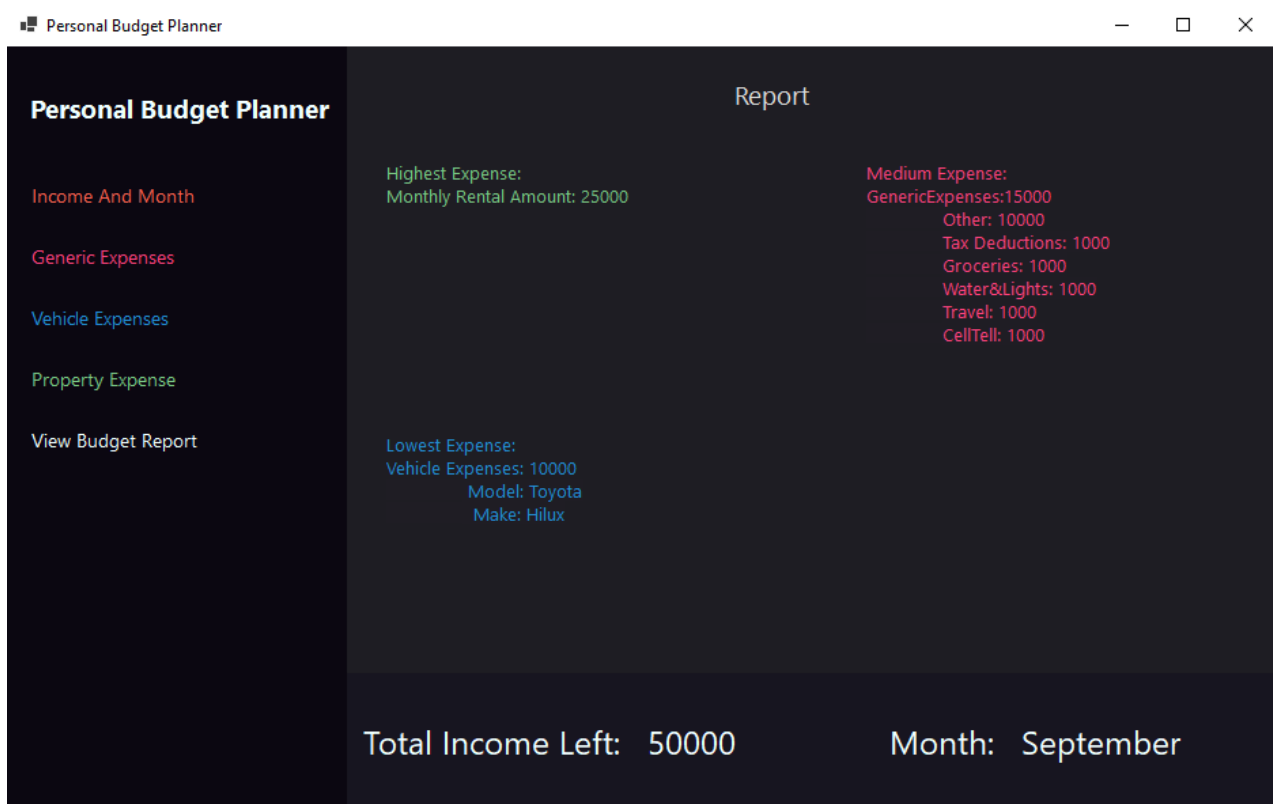
Delete

Total Income Left: 5040

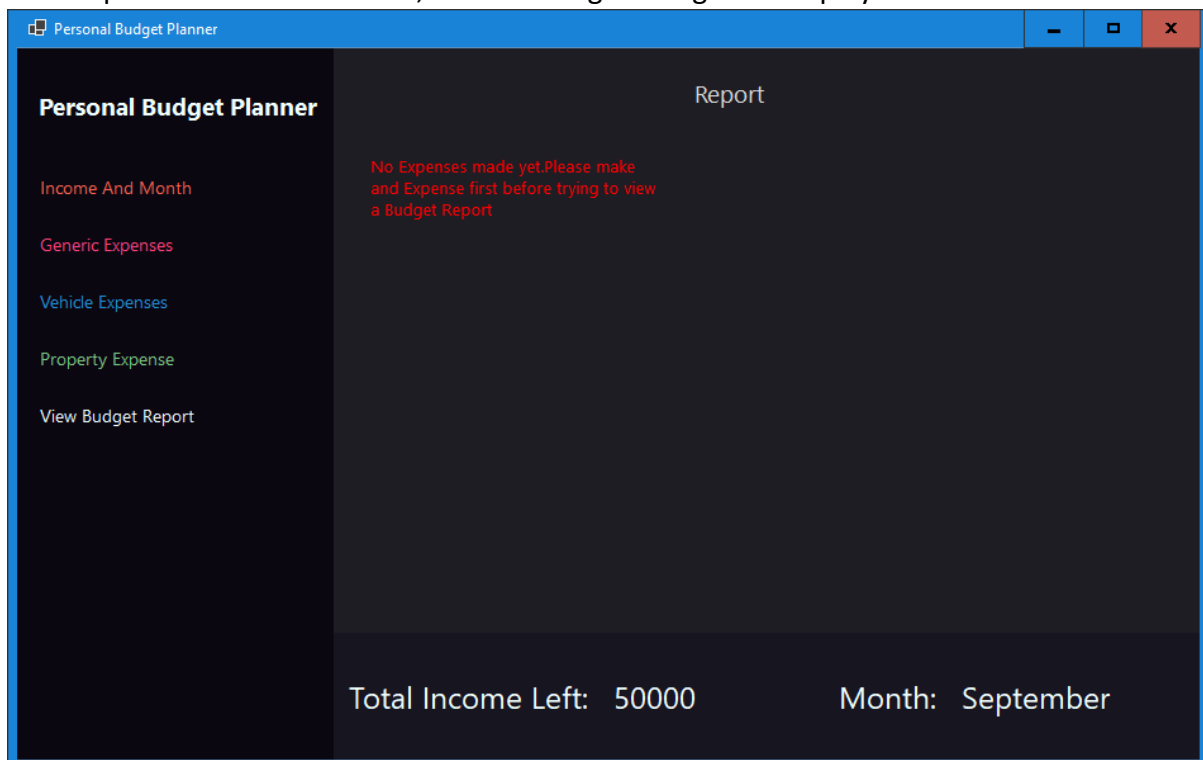
Month: April

Note that once an expense is made the total Income left is updated and the “Delete” button becomes available to delete that specific expense. Also note that if you click another button and leave the Generic Expenses screen, you can click back on “Generic Expense” and it will display the latest Generic Expense saved. This is true for all expenses.

Once an expense is made you can access the “View Budget Report” to view all expenses made, in descending order. Here all expense types have been made for a budget of R100 000 in September



If no expenses have been made, the following message will display



Changes made based on feedback

In the original version of this application (Task 1). For “: Home loan calculations and warning implemented correctly.” The Home Loan calculation was incorrect. This result to multiple things being thrown off. This has now been fixed and the home Loan display the correct amount using simple interest as stipulated by the POE for task 1. For “: Available monthly money calculated correctly and displayed”, the incorrect Home Loan calculation was throwing this output off. Now that it has been fixed and the money available is correctly calculated and displayed. Inheritance was slightly upside down. For “Application makes use of classes and inheritance in a logical way”, The classes have been adjusted and this has been fixed. For “: The expenses are stored in an array”, this was altered and the expenses stored in a list but correctly with a list handler class. The expense have now been stored correctly in a generic collection, in the form of a List, with a list handler class being made to correctly work with the list. For “Code is well structured and documented”, there was some issues with not renaming component, untidy methods to clear textboxes and check null inputs and there was not enough comments to explain the logic of the application correctly. These have all been fixed. All names are appropriate, methods have been optimised and ample comments given to explain logic. For extra marks, the application has been improved using heavily advanced GUI.

Support

For support, please contact 20108381@vcconnect.co.za