Maryam Gadiali

Design of the Application

I created this GUI using Python 3, with the tkinter module. This was my first time creating a GUI, as I started off this year as a beginner in programming, so I had to learn the various features of the module through websites and YouTube tutorials (majority was learnt from the channel: “freeCodeCamp”).

I used checkboxes, radio buttons, drop menus, buttons, labels, frames, windows, error boxes and entry boxes which provided the main backbone for this program.

The checkboxes were used for when a user would have the choice to select multiple formulas to calculate, without being restricted to calculating only one formula at a time.

The Radio buttons were used for singular selection, such as with the Present and the Future value equations in Tab 2, as the variables entered such as the choice for amount, has 2 different meanings when it comes to each equation.

The drop menu was used for singular selection as well, for Tab 1, where the user had to select which class of formulas they wanted to proceed with. I used this to avoid repeating radio buttons, which provides more variability in my program.

A lot of buttons were used throughout to increase interactivity, open and close various windows, frames, error boxes as well as used to initialise a loop again if an error was made, and more. The buttons also had to be disabled constantly throughout the program, after the user had made their selections, to avoid overlapping results, which ruins the layout of the interface, and would make it confusing to users.

Labels were used to convey messages across, without the need for the user to interact directly with the system and were used to show what values a person needed to enter for the equation as well as displaying the final results, and more.

Frames were used to help support the organisation of the program, and make it look more neater, which enhances user experience. I also contained widgets within the frames, so it would be easier to destroy, and have it restarted if there was a value error, where the wrong types were given.

I used 2 named windows, one was called 'Root', and the other was called 'Top'.

Root served as the main window throughout the program, where it was used for the beginning of Tab 1, and for the entirety of Tab 2 and Tab 3. 'Top' window was used for Tab 1 after the user had chosen which formulas they wanted to calculate, in order to isolate the entry values screen from the selection features of the root window, in order to provide the user with a main focus, which was to enter the values on the new window. I also had to keep using the lift feature, to ensure users would have direct access to the current window of focus, instead of having that window hiding behind other windows.

I used error boxes to account for when users would enter no values, or did not enter an integer or float value, this was presented as a pop-up box, where then the user would be sent back to a certain tab or frame, so they can reenter the values correctly.

Entry boxes were used primarily for the user to enter the necessary values, no entries had been preset, as there was no need.

As well as using the tkinter module, I also utilised the Numpy Financial Module, where I used the Net Present Value and Internal Rate of Return functions to calculate their formulas.
I imported the os and sys modules, as I used them to reset the entire program, which makes the user go back to Tab 1.

Code Fragments and Explanation

Lines 11 to 26 creates the root window, sets it so the window is brought to the front of the screen, as well as determines the size of the window relative to the screen size.
Lines 1094 to 1097 creates two navigation buttons, which swaps over to the next and last tab, and this is a feature in all the tabs. If the tab is 1, then the 'last tab' button is disabled, and if the tab is 3, the 'next tab' button is disabled, this is to make it clear to the user that there are only 3 tabs available, and that not being able to move to the next tab is not a bug.
Lines 1099 to 1110 initialises the options for the drop-down list, and creates a button, where after the formula class type has been selected, then the user can select their formulas.
Lines 726 to 786 disables the previous button used, so results don't overlap, and then there are 4 conditional statements based on which formula class type had been selected in the last step, so checkboxes can be presented with their respective formulas, where the user can choose which formulas they wish to calculate, and then they click a button where a new function is called for each 4 formula class types.
When the new function is called, a new window called 'top' is created, so users can enter values for the next step in there.
Lines 31 to 233 covers the Profitability class, and considers which individual formulas had been selected, whereby if it has been selected, users can enter the individual values into entry boxes, and press a button when they are done, where a result frame will open displaying the results for each formula. There is also a button to returning to the initial process of Tab 1 in the root window. If the entry was 0 or not an integer or float number, then an error message box appears telling them to enter a number or decimal, and then they are provided a button to restart that window, so they can enter the correct values in this time.

Lines 236 to 430 covers the Liquidity class, Lines 436 to 576 covers the Leverage class, and Lines 582 to 723 covers the Operating class, and they all follow the same procedure as the Profitability class.
After the users have received their results in the 'top' window for their formulas, they can return to Tab 1 using the button at the top of the window.

Lines 797-814 and 1019-1092 contains the coding related to Tab 2.
First, it clears the widgets of the previous window and resizes the window suited for tab 2.
It provides 3 entry boxes, with labels describing each box besides it. These 3 boxes are for the variables relating to the Present and Future value formulas. After the user clicks the button to confirm selection, the user can select from two radio buttons which formula they want to calculate. After they have confirmed their selection by clicking the button, the result is displayed below. If the wrong or no values are given for the entry boxes, an error box appears, resetting

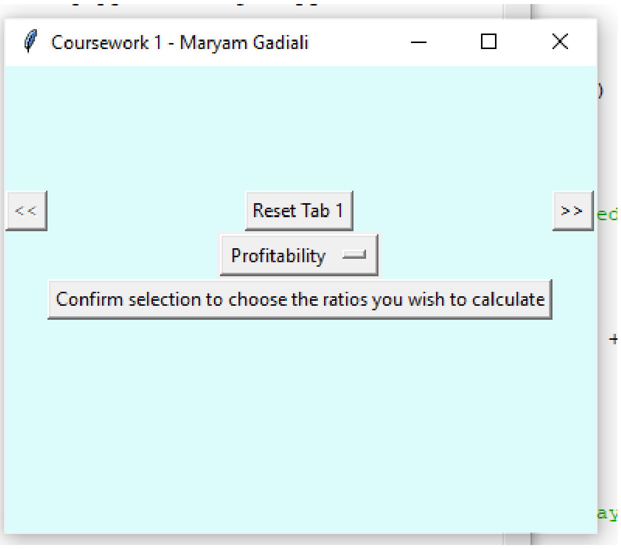
the tab page, so they can enter the correct values. Buttons were again disabled after user selection, to avoid results overlapping and ruining the design and layout of the interface.

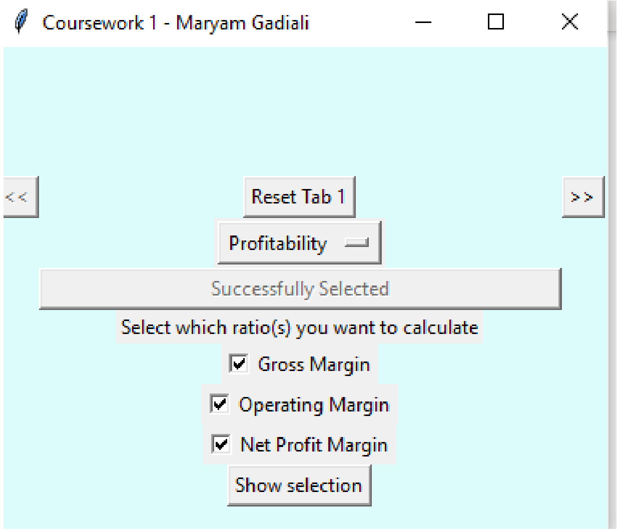
Lines 816 to 1017 contain the coding for Tab 3. First the code sets the size for the root window and displays 2 entry boxes with labels for the user to enter the initial investment and the discount rate. After the user confirms the values with a button, a table frame for entering cash flows for a 5-year period appears. This gets rid of the last frame, which consisted of the 2 entry boxes above, in order to utilise space so in the later steps, the interface doesn't go off the screen. There is an entry box to enter each cash flow, and after the completion button is pressed, an option frame appears, presenting checkboxes for which formulas they want to calculate. This gets rid of the table frame, so to save screen space. After the selection has been confirmed, the results are displayed below. For each of the multiple inputs in this tab, there is an error pop up box where if no selections/values are made or the wrong ones are given, the user is taken to the start of the tab 3 process.
All the tabs have a button to reset the current tab they are in.

A notable feature to mention that was used heavily in the python code was the use of functions, as there were a lot of different steps involved, so functions had to be used to segment and block the code. Global variables were also used because of the functions, so they could be used throughout the program as a lot of the different steps carry the same variables as a different step.
Assumptions made are all the formulas I've used throughout, which were largely sourced from the lecturer's slides, as well as the method I used to calculate the Payback Period, where I calculated the cumulative values, and if it recovered more than the initial investment, I would use the remainder between the initial investment and the last cumulative value before that period, and divide it by the total value of this period, which assumes that the values for that period is gained at a constant rate over the year which is how the number of days are calculated, by multiplying the result by 365.

Screenshot of the Application (UI and output)

Root Window Tab 1:


Coursework 1 - Maryam Gadiali
< <
Reset Tab 1
> >
Profitability
Confirm selection to choose the ratios you wish to calculate



Successfully Selected

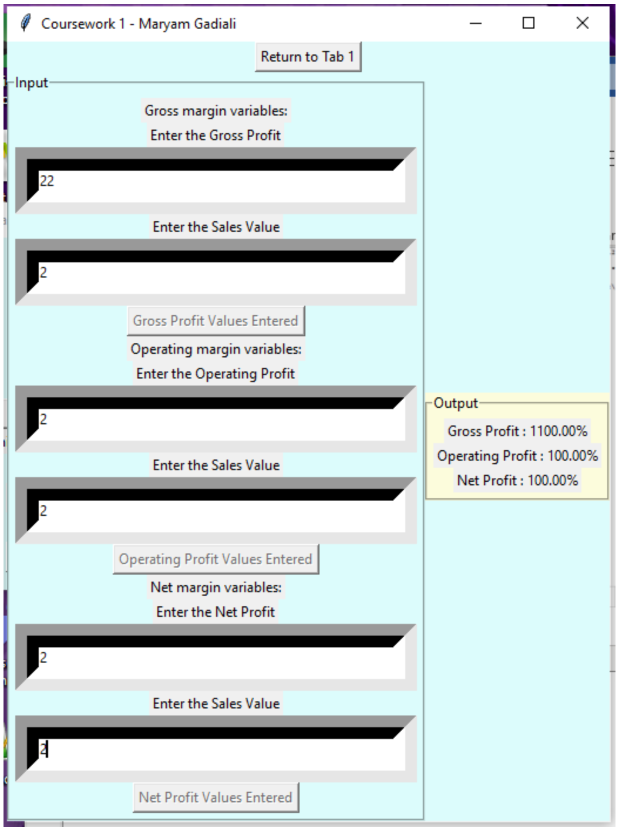
Select which ratio(s) you want to calculate

☑ Gross Margin

☑ Operating Margin

☑ Net Profit Margin

Top Window Tab 1:



Gross margin variables:

Enter the Gross Profit: 22

Enter the Sales Value: 2

Gross Profit Values Entered

Operating margin variables:

Enter the Operating Profit: 2

Enter the Sales Value: 2

Operating Profit Values Entered

Net margin variables:

Enter the Net Profit: 2

Enter the Sales Value: 2

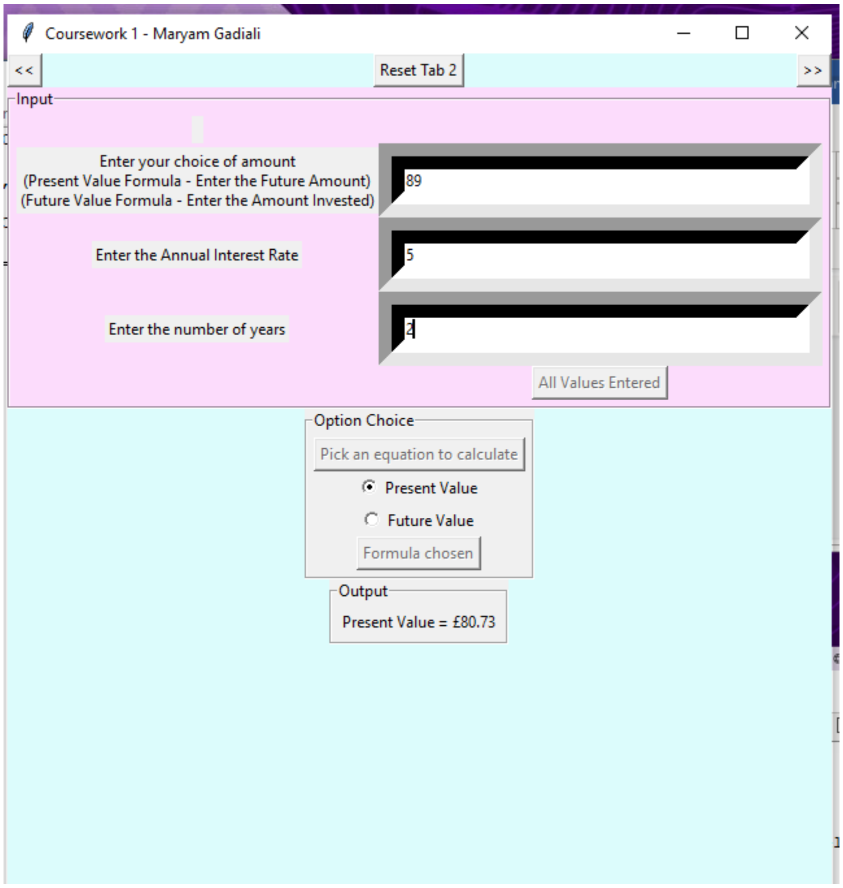
Net Profit Values Entered

Gross Profit : 1100.00%

Operating Profit : 100.00%

Net Profit : 100.00%

Tab 2:



Enter your choice of amount
(Present Value Formula - Enter the Future Amount)
(Future Value Formula - Enter the Amount Invested): 89

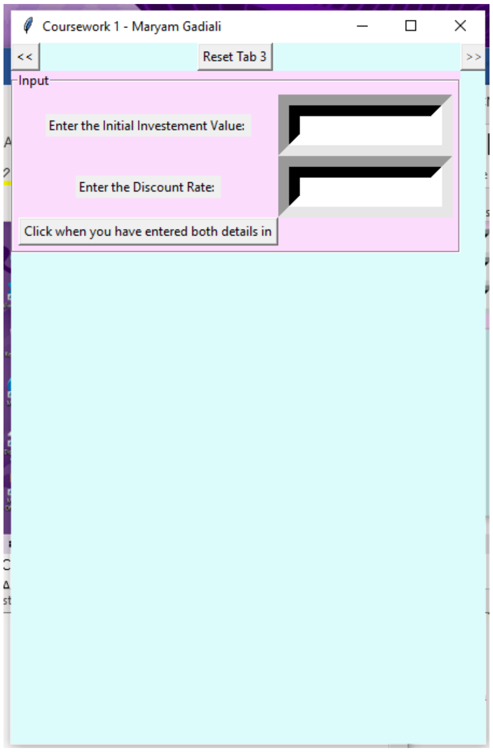
Enter the Annual Interest Rate: 5

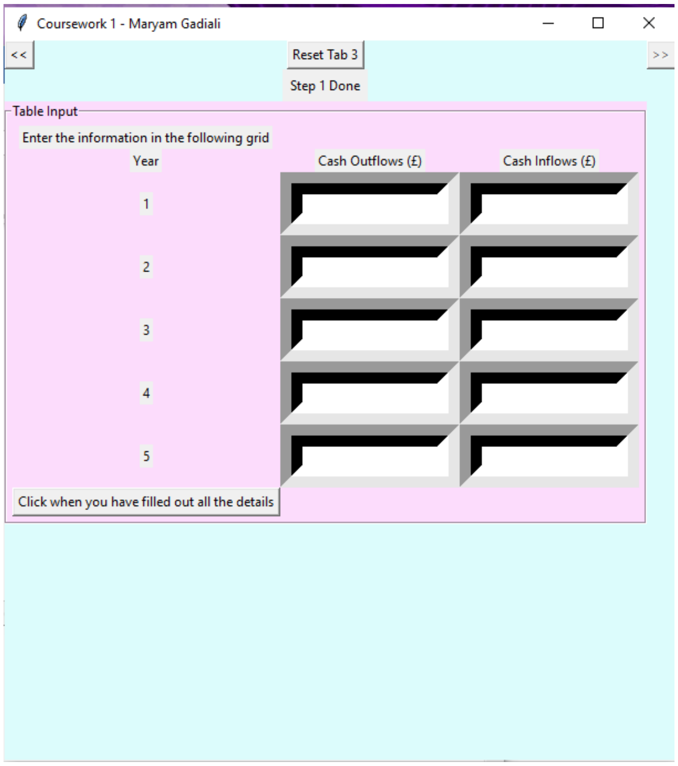
Enter the number of years: 2

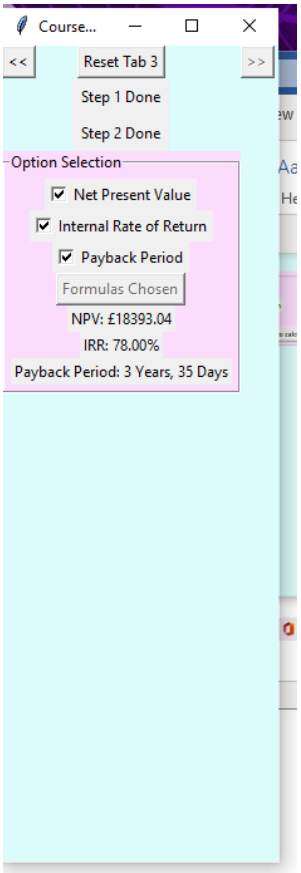
● Present Value
○ Future Value

Present Value = £80.73

Tab 3 step by step:


Coursework 1 - Maryam Gadiali
<<
Reset Tab 3
>>
Input
Enter the Initial Investement Value:
Enter the Discount Rate:
Click when you have entered both details in


Coursework 1 - Maryam Gadiali
<<
Reset Tab 3
>>
Step 1 Done
Table Input
Enter the information in the following grid
Year
Cash Outflows (£)
Cash Inflows (£)
1
2
3
4
5
Click when you have filled out all the details



☑ Net Present Value

☑ Internal Rate of Return

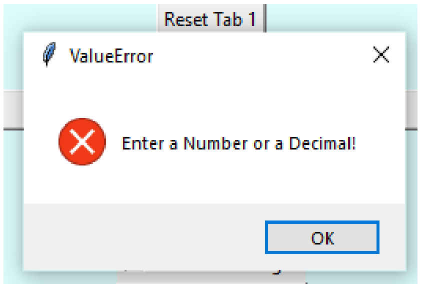
☑ Payback Period

NPV: £18393.04

IRR: 78.00%

Payback Period: 3 Years, 35 Days

Typical design for an Error box:



Enter a Number or a Decimal!

Future enhancements for the Application

In the future, if I had more time for this, I would focus on exploring more design features to make the user experience better, as well as finding a way for dynamically changing the size of the window throughout, so I would not have to give it preset values most of the time. I would also attempt finding shorter ways to code this program whilst achieving the same result, which would rely on experimenting with more functions.