COM5002

Personal Finance Manager

Wadzanai Chawatama

Yeshiva University

215 Lexington Ave

wchawata@mail.yu.edu

Yehuda Wexler

Yeshiva University

215 Lexington Ave

ywexler@mail.yu.edu

Project Description:

A financial planning application in which users can upload their monthly income and expenses- and effectively plan for retirement at 65. This application will allow users to add income and expenses to determine monthly progress towards a predetermined retirement goal. To achieve this, we propose utilizing dictionaries to store userID:Age,Target funds at retirement. As well as utilizing Arrays or linked lists to store income and expenses- using insert/remove to add and edit or remove items. We will also include a retirement calculator to allow users to determine how far from retirement they are..

Significance:

This application will provide us with an avenue to explore implementing common algorithms such as insert and remove, as well as extensive practice with

dictionaries and linked lists. Building this application will allow us to work on skills that will serve us throughout our careers moving forward.

Project Specification and Design

A financial planning application that allows users to input their income and expenses and track their progress towards a retirement goal. This also allows users to effectively budget for large expenses and /or emergency funds when necessary.

Variables

Variable	Explanation
(F) MonthlyIncome	Total Monthly Income
(List)incomeSources	List of sources the income comes from
(F)totalExpenses	Sum of expenses added by user
(List)expenses	Names of expenses
(INT)YearstoRetirement	Number of yearsto hit retirement goal at current rate
(F) currentSavings	Starting point-money already saved
(F) Net Change	Difference to be assigned to the Current_savings

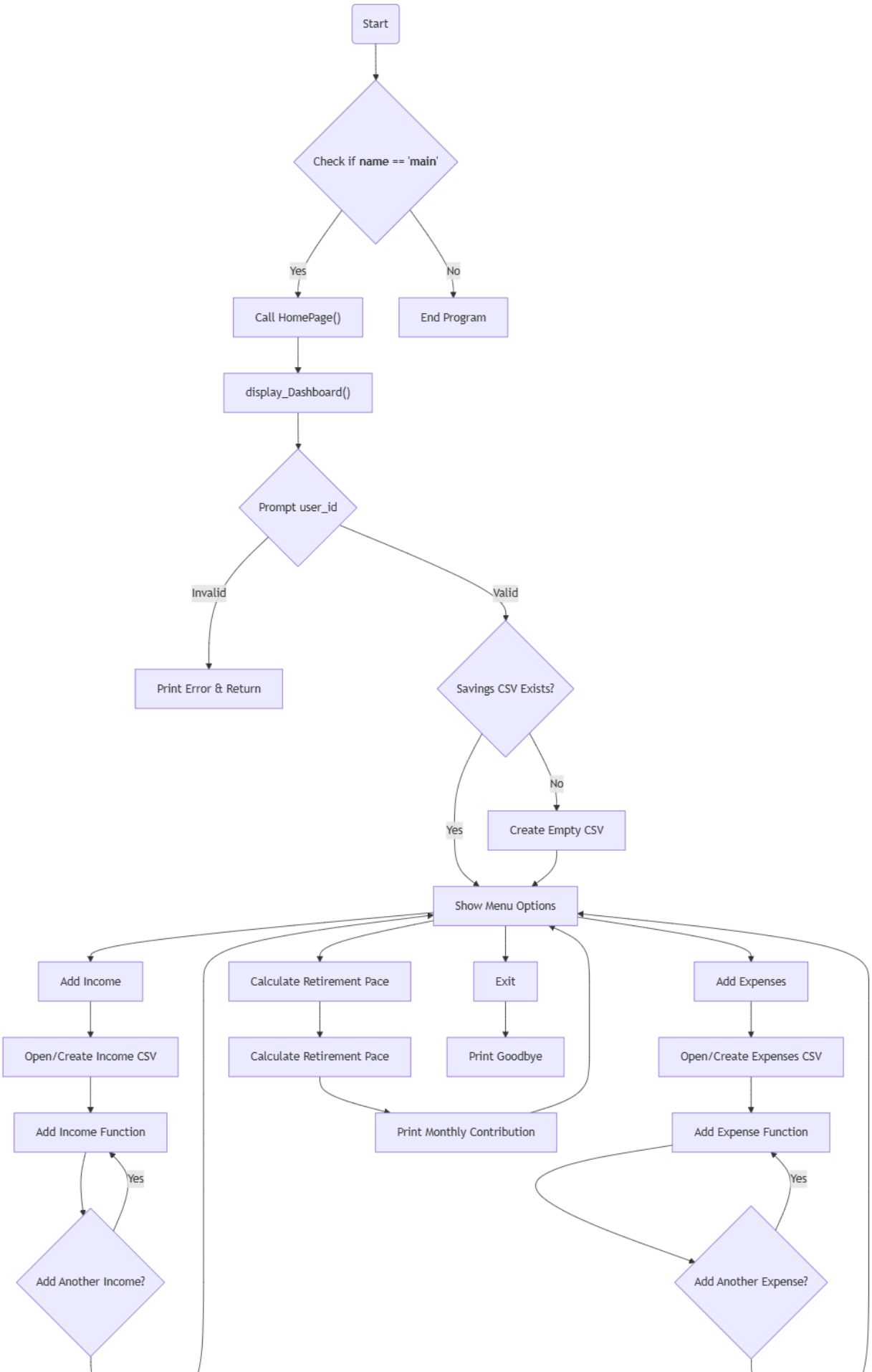
Project Flowchart

Updated Flowchart: Upon calling the program (main) the Homepage is opened, displaying the dashboard. This first asks for the user_id to identity the proper CSV's to utilize (or create a new set).

Error handling on this step dictates that user_id must be an integer.

Once the system has identified the proper CSV's to use, it offers the user a menu. Income, Expenses, RetirementCalc, Exit.

The income pathway opens/ uses the income CSV to add inputs (including timestamp for later historical data) and calls the Update_Savings function to send the information the savings CSV as well.


Start
Check if name == 'main'
Yes
No
Call HomePage()
End Program
display_Dashboard()
Prompt user_id
Invalid
Valid
Print Error & Return
Savings CSV Exists?
No
Yes
Create Empty CSV
Show Menu Options
Add Income
Calculate Retirement Pace
Exit
Add Expenses
Open/Create Income CSV
Calculate Retirement Pace
Print Goodbye
Open/Create Expenses CSV
Add Income Function
Print Monthly Contribution
Add Expense Function
Yes
Yes
Add Another Income?
Add Another Expense?

Data Acquisition

Data Sourced by user inputs. We will begin with a preset user who will be 25 years old with income of $3,000 and average expenses of $1,800. Resulting in a savings rate of $1,200 per month. User would like to retire with savings of $2.5MM (in line with a 4% safe drawdown rate).

Current Progress and Implementation

The project has made significant progress in implementing a financial planning application. The core functionality has been developed using Python, with the following key components implemented:

1. Data storage: The program uses CSV files to store user data, including income, expenses, and savings information.
2. User interface: A command-line interface has been created, allowing users to interact with the application through a simple menu system.
3. Income tracking: Users can add income entries, specifying the amount and source.
4. Expense tracking: Users can add expense entries, specifying the amount and category.
5. Retirement pace calculation: The application can calculate the monthly savings required to reach a retirement goal based on the user's current age and savings.
6. Data persistence: User data is saved to CSV files, allowing for data retention between sessions.

Discussions

During the implementation of this project, we encountered several challenges:

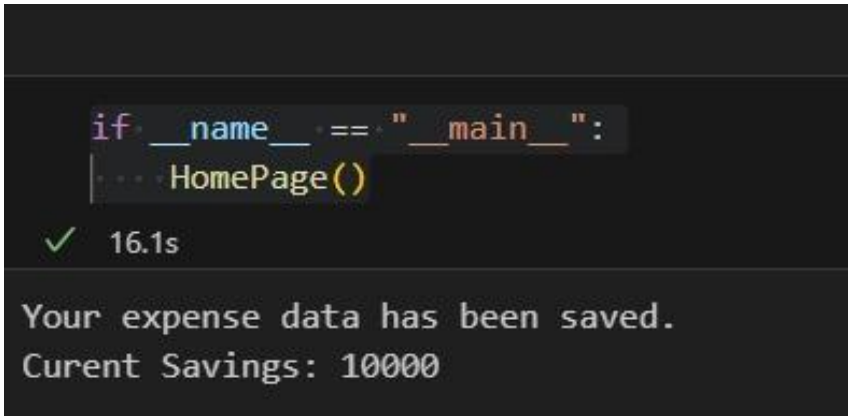
1. Data persistence: Initially, there was difficulty in efficiently storing and retrieving user data. This was solved by implementing a CSV-based storage system, which allows for easy data manipulation and retrieval.
2. Getting the income and expenses to correctly reflect in the updated savings. There were some issues with creating a new CSV and not remembering the current savings. This was adjusted by creating a separate function for updating savings.
3. While trying to fix the savings issue we managed to break everything :) it took a few hours but after some debugging we were able to recreate the code in a more efficient and robust manner.

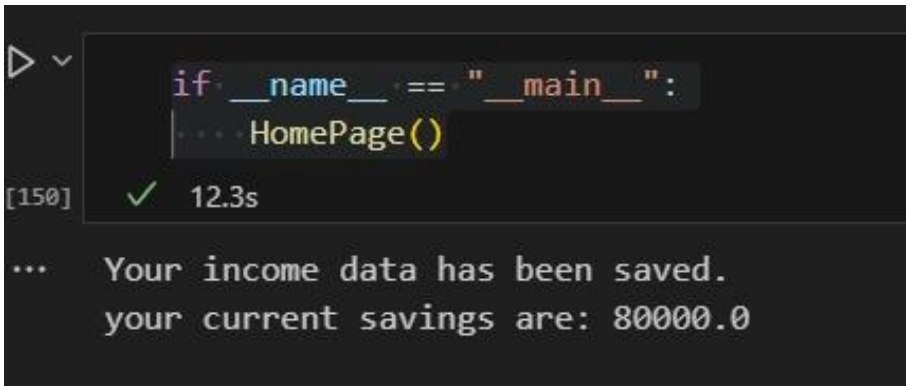
Roadmap

1. General User Interface- We would like to implement a simple GUI for the program.
2. Retirement Calculator- We would like to implement a more robust retirement pace calculator. Currently, the calculator only returns the amount that needs to be saved monthly to achieve retirement goal by 65.

timestamp	expense	category
3/25/2025 14:20	10000	car
3/25/2025 14:24	300	Metrocard
3/25/2025 14:24	20	food
3/25/2025 14:24	120000	tuition

	A	B	C
1	timestamp	income	source
2	3/25/2025 14:18	200	me
3	3/25/2025 14:22	70000	lottery
4			
5			





Project Timeline

Task (Schedule your own tasks below. Adjust it as time goes if needed.)	Time (Schedule your own time below. Adjust it as time goes if needed.)
Read related resources and write project proposal.	2/13 ~ 2/26

Implement the structure and skeleton code	2/27-3/12
Implement Add income and Expense- write progress report	3/13-3/26
Implement more robust retirement calculation	3/28-4/3
Create GUI	4/4-4/10
Test Code	4/11-4/14
Prepare final submission and PPT	4/15-5/5

References:

Claude. (2025, February 24). Personal communication [Online conversation].

Claude. (2025, February 16). Personal communication [Online conversation].