

COM5002

- WADZANAI CHAWATAMA
- YESHIVA UNIVERSITY
- 215 LEXINGTON AVE
- WCHAWATA@MAIL.YU.EDU

- YEHUDA WEXLER
- YESHIVA UNIVERSITY
- 215 LEXINGTON AVE
- YWEXLER@MAIL.YU.EDU

PROJECT DESCRIPTION:

Develop a financial planning application to enable users to track income, expenses, and progress towards retirement.

- Key Features:
 - Upload and track income and expenses.
 - View progress towards retirement goals.

SIGNIFICANCE:

This project provides hands-on experience with:

- Key data structures:
- Algorithm implementation
- Practical financial planning knowledge

Description

Financial Planning Application which:

- Intakes Income and Expense Data
- Updates Current Savings
- Calculates Pace to Retirement
- Stores data in CSV



System Design

User Interface:

Home page with four options:

- Add Income
- Add Expense
- Pace to Retirement
- Exit



Backend Design

User data stored in CSV
Simple retirement
calculator



System Breakdown

When adding Income:

First find the income and savings CSVs, if the CSVs don't exist, create new ones.

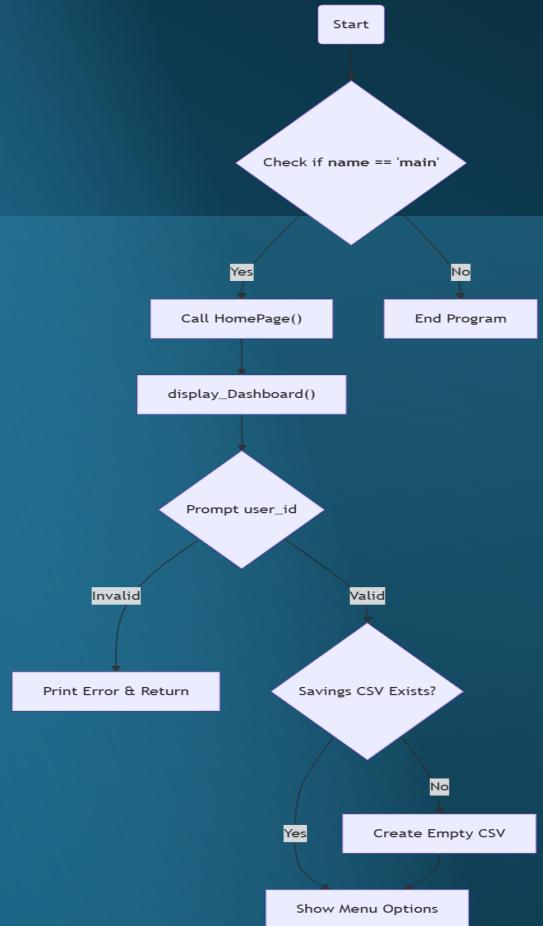
Given the CSVs exist, pull in the current savings (nonlocal variable) and add income and source.

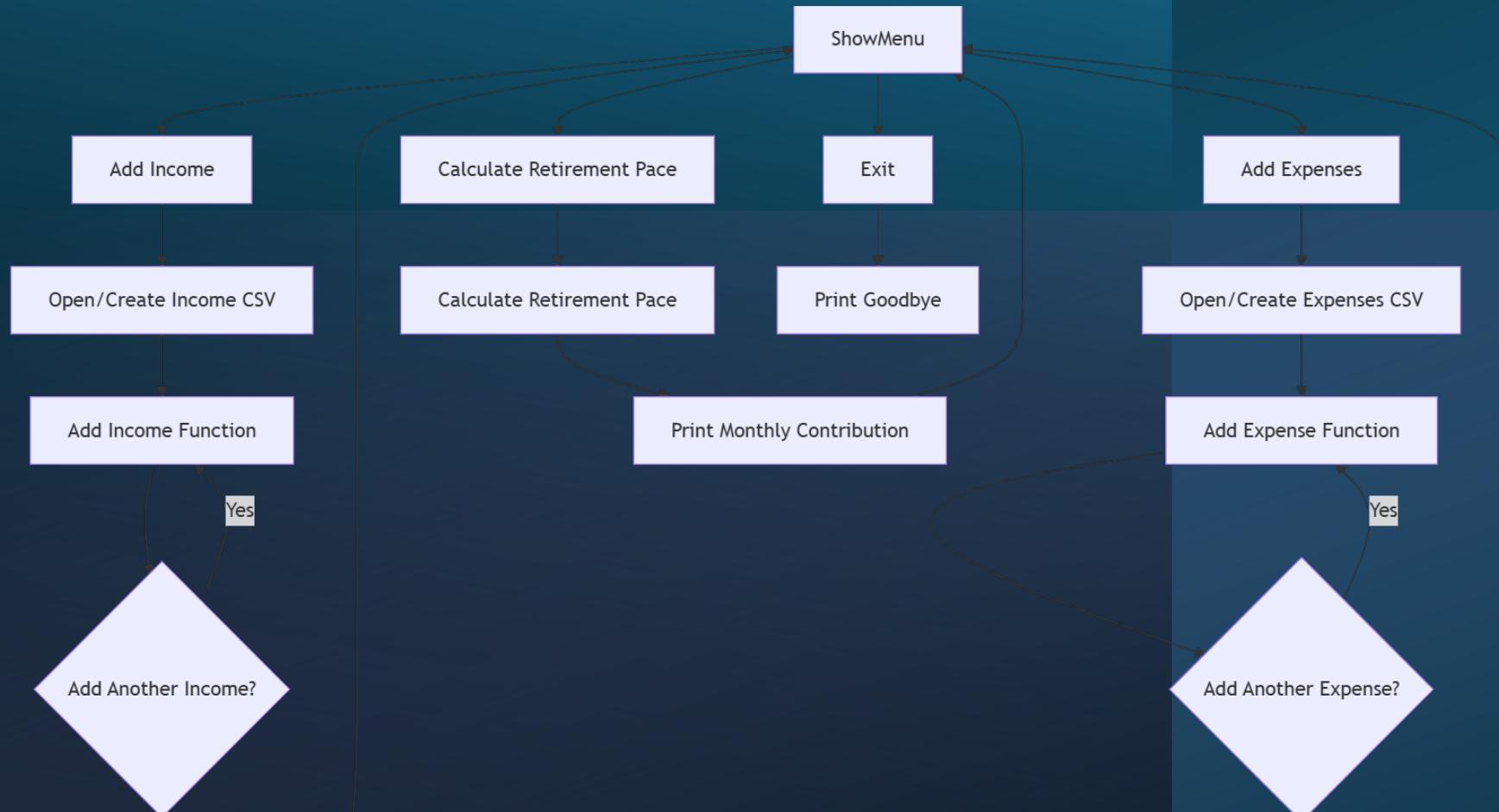
Create a df for new_income and concatenate it to the income DF

Repeat this process for savings

Timestamps are attached for historical data tracking.
And the user is prompted to add another entry or exit.







Current Progress and Implementation

- 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.
-
- 1. Expense tracking: Users can add expense entries, specifying the amount and category.
 - 2. 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.
 - 3. Data persistence: User data is saved to CSV files, allowing for data retention between sessions.

Current progress and Design

11

```
if __name__ == "__main__":
    HomePage()
✓ 16.1s
```

Your expense data has been saved.
Current Savings: 10000

	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			
6			

```
if __name__ == "__main__":
    HomePage()
[150] ✓ 12.3s
```

... Your income data has been saved.
your current savings are: 80000.0

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

Discussion

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

General User Interface- We would like to implement a simple GUI for the program.



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.



The background of the image features a repeating pattern of speech bubbles in various colors, including yellow, red, purple, and grey, all containing a large blue question mark. This pattern covers the entire slide.

Any Questions?