

# MIS 20080 – Group 10 Project

## USER MANUAL

SAM URELL – 22465384

ARNAV RATTAN – 22204232

HUGH MUNRO – 22347541

## Table of Contents

<b>1.Contributions</b>	<b>2</b>
1. Sam Urell	2
2. Arnav Rattan	2
3. Hugh Munro	2
<b>2.User Manual</b>	<b>2</b>
1. Introduction	2
2. Getting Started!	2
2.1 Installation	2
2.2 Accessing the Application	3
3. User Interface Overview	3
3.1 Navigation	3
3.2 Key Features	3
4. Budget Calculator	3
5. Savings and Investments Calculator	3
<b>3. UML Activity Diagrams</b>	<b>4</b>
1. Budget Calculator	4
2. Savings and Investments	5
2.1 Data Fetching Function	5
2.2 Savings Function	5
2.3 Growth Function	6
2.4 Output Function	6

## Contributions

1. **Sam Urell:** I conceived the idea of budget calculator. I also created and managed the GitHub repository allowing me and my team to easily collaborate on our code and project ideas. I designed and created the budget.py module, along with Homepage and Budget Viewer page with a strong emphasis towards intuitiveness and functionality. I ensured optimal reusability and organisation within my code through my utilisation of the Object-Oriented Programming paradigm. Finally, I designed the UML diagram relating to Budget Calculator.
2. **Arnav Rattan:** I was responsible for coming up with the idea for the Savings and Investments Calculator. Henceforth, I researched and created the web-page piece by piece. I also learnt how to use GitHub and consistently updated my code for a better collaborative experience. This also allowed me to learn about version control and keeping track of changes over time. Responsible for commenting on the code and creating docstrings for the functions. I created all the functions and learnt how to communicate with an API. I also designed and created all the UML diagrams relating to the Investment Calculator. Created the user manual for the savings and investments calculator. Helped in pathing of the pictures for the Team member web page. Ensured code robustness and clarity by testing different user inputs. Also organised group meetings and delegated tasks effectively.
3. **Hugh Munro:** For my part I predominantly contributed to the frontend side of the project with specific focus on refining the User Interface (UI). My efforts were mainly directed towards the Main Page and Team Members page, where I implemented some user-friendly features and clean visuals. I also worked alongside my teammates, Sam and Arnav, helping them with design and functional choices, stressing user-friendliness, and correcting syntax and spelling errors. Additionally, I took charge of the task of creating and compiling the User Manual ensuring cohesive formatting and structured sections through editing and redesigning.

## User Manual

### 1. Introduction

Our Streamlit application allows users to calculate and store their budget for any given time horizon and offers risk-tailored investment suggestions to help users reach their savings goals.

### 2. Getting Started!

2.1 Installation: To get started with this application you must ensure the following Python libraries are imported:

- [streamlit](#): A powerful framework for creating web applications with minimal effort.
- [pandas](#): A versatile data manipulation library that provides data structures for efficiently storing and analysing data.
- [os](#): A module providing a way to interact with the operating system, facilitating tasks such as file and directory manipulation.
- [yfinance](#): A package for retrieving financial data from Yahoo Finance.

- [pillow](#): imaging library that adds image processing capabilities.
- [matplotlib](#): library for creating powerful data visualisations.
- [json](#): A module for encoding and decoding JSON data.
- [datetime](#): A module for working with dates and times.

In order to install streamlit, pandas, matplotlib, pillow, and yfinance you can simply enter the following command in your terminal or command prompt: **pip install [insert library]**.

The remaining packages (os, json, and datetime) should already be included in Python by default. For additional information regarding the installation of the dependencies, click one of the hyperlinks above.

2.2 Accessing the application: To access the application, you must first download the zip file *MIS\_20080\_Group\_10* and set it as the working directory. Finally run main.py using the command “**streamlit run main.py**” in your terminal or command prompt.

### 3. User Interface Overview

3.1 Navigation: Upon launching the application, you will be greeted with the main page. From there you can navigate the web application using the sidebar. All web pages can be conveniently found on the sidebar and can be accessed at all times by clicking the arrow in the top-left hand corner revealing the sidebar. This allows the user to seamlessly switch between the diverse features of the application with ease.

3.2 Key Features: Within our application we’ve sought to make the financial planning experience as intuitive and simple at every step of the way. The selection of clean and informative graphs and visuals help the user to gain immediate at-a-glance insights into their financial position allowing for enhanced understanding and clarity. Additionally, the strategic integration of intuitive selection boxes, widgets, and prompts help to guide the user through the platform smoothly with a sense of confidence and peace of mind.

### 4. Budget Calculator

The Budget Creator allows users to create and store comprehensive 12-month budgets. The user has the flexibility to create a new budget beginning from the current date or import a pre-existing budget from a .json file. This budget can contain cash flows of varying amounts and bases, for example, daily, weekly, monthly, or one-time, over varying time periods. This budget can then be exported to a .json file for future use/reference. The Budget Viewer tab allows the user to view a detailed breakdown of inflows and outflows over a specific time period, along with a snapshot of the user’s current checking account balance over the budget period. Finally, within the Savings and Investment Calculator, the user has the option to add a monthly outflow of savings on high and low-risk investments directly into the current budget.

### 5. Savings and Investment Calculator

This simple and intuitive calculator allows users to easily plan for future funds and optimally invest their savings tailored to their risk tolerance. The calculator prompts users to input basic information

such as the time horizon and fund amount. The user should then define their risk preference and the application provides several different investment suggestions.

For more risk averse users, the Lower Risk option provides a selection of three US government bonds in addition to their respective yield curves (Please note that in a well-functioning economy, the yield for long term bonds should be higher than the short-term bonds due to interest rate risk however our investment suggestions may show an inverted yield curve due to current market conditions).

For Higher Risk tolerant users, a basic line chart of the S&P 500 is graphed while a slider for returns is provided to allow the user to set their desired rate of return, enabling the user flexibility in managing their level of risk associated with equities.

The program then calculates the required monthly investment amount by computing the annuity amount from the future value of an annuity formula.

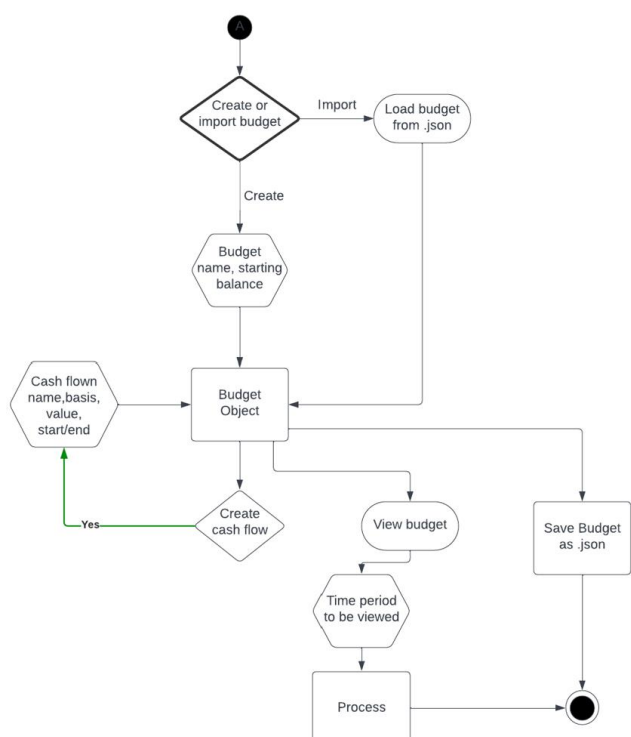
$$FV = M((1 + r)^n - 1)/r$$

This data is then visualised through a series of graphs to show the power of compound interest.

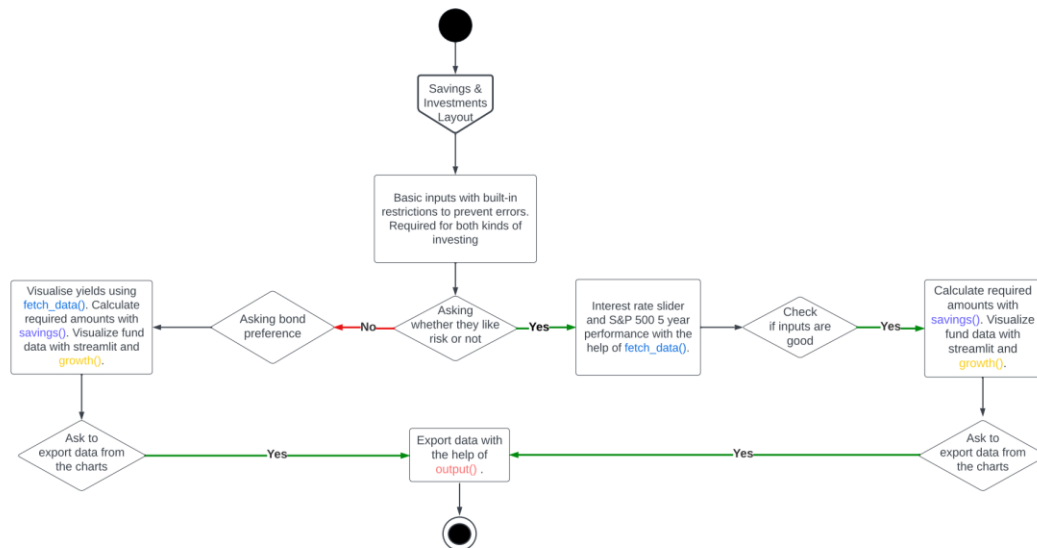
At the end of the page the user is instructed to select the format they would like to export the generated table. Users are advised to select the format type based on their desired use of the data. Excel and CSV files are easily read by humans while JSON is particularly useful when for inter-communicating information between programs or over the web.

## UML Activity Diagrams

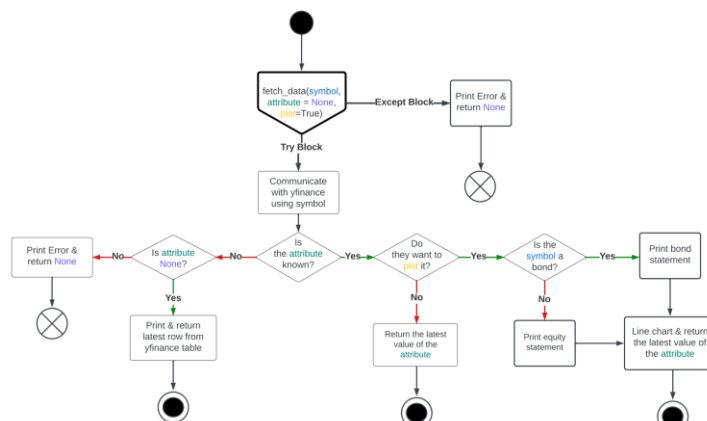
### 1. Budget Calculator:



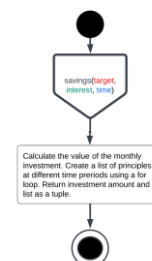
## 2. Savings and Investments:



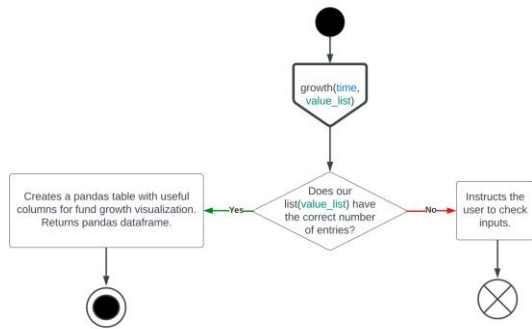
### 2.1 Data Fetching Function:



### 2.2 Savings Function:



## 2.3 Growth Function:



## 2.4 Output Function:

