

**Midterm Project Report**

**Advanced Computer Programming**

**Web Services with Python**

**Student Name**  **: HARSHADA SHANKARRAO**

**MANEKAR**

**Student ID : 112021190**

**Teacher**  **: DINH-TRUNG VU**

**2024-04**

**Chapter 1 - Introduction**

**1.1 GitHub**

1) **Personal GitHub Account:** 112021190 (Harshada Manekar)

2) **Group GitHub Account:** Fullymed

3) **Group Project Repository:** Amigo

4) **List of Submitted Files:** 112021190-harshada.py

**1.2 Topic**

Our project is dedicated to developing web services tailored for financial tracking, aiming to streamline record-keeping and goal setting processes for seamless personal finance management. Rooted in the ambition of empowering users to monitor their financial transactions meticulously while setting clear objectives for savings and expenditures, my role focuses on providing tailored savings recommendations aligned with individual financial profiles. Leveraging sophisticated algorithms and data analysis techniques, our savings recommendation feature equips users with bespoke strategies to optimize their savings pursuits, fostering a culture of financial empowerment. Coupled with a meticulously crafted user experience, our platform exemplifies principles of code organization and reusability, resonating with our dedication to excellence and fostering financial literacy and responsibility within our community.

**1.3 Project Overview**

The project aims to develop a web service for financial tracking, enabling users to record their finances, set savings targets, and track expenses. The purpose is to provide users with a convenient platform to manage their finances effectively. Savings Recommendations: Within the Finance Tracker web service, one of the key features is savings recommendations. This component focuses on providing users with personalized recommendations for allocating their income towards savings, needs, and wants. Importance: The savings recommendations feature plays a crucial role in helping users achieve their financial goals. By offering tailored suggestions based on the user's income and financial situation, it assists in optimizing budget allocation and promoting responsible financial management.

**Chapter 2 - Implementation**

### **2.1 Class**

The “Savings Recommendation” class is a fundamental component of our financial tool, responsible for computing budget allocation based on the user's income. By encapsulating the logic for this crucial task, the class enables users to gain insights into their spending habits and make informed financial decisions.

Features:

**User Input Handling**:The code employs a robust mechanism for soliciting user input, ensuring a smooth interaction with the program.It validates user input meticulously, utilizing regular expressions to confirm that inputs are valid numerical values.

**Budget Allocation Calculation**:

Utilizing the information provided by the user, the code performs precise calculations to determine the appropriate allocation of funds for taxes, needs, and wants.It employs accurate mathematical formulas to compute the exact amounts earmarked for each category, thereby fostering financial transparency and comprehension.

**Menu-driven Interface**:The program features an intuitive menu-driven interface, enhancing user experience by offering clear and structured options.

Through this interface, users can effortlessly navigate the program's functionalities, promoting accessibility and ease of use.

**Data Storage to CSV**:

To facilitate data management and archival, the code allows users to seamlessly store the calculated financial data into a CSV file.

Leveraging the **csv** module, the program ensures the integrity and organization of the stored data, aligning with best practices in data storage and handling.

**Object-oriented Approach**:

Embracing the principles of object-oriented programming, the code embodies a modular and extensible design.

By encapsulating related functionalities within classes (**SavingsRecommendation** and **StoreData**), the code promotes code reusability, maintainability, and scalability.

**Reusability and Extensibility**:

Through its structured and modular architecture, the code lays a foundation for reusability and extensibility, paving the way for future enhancements and integrations.

Components such as the **SavingsRecommendation** class are designed with flexibility in mind, facilitating seamless integration into other projects or potential expansion with additional functionalities.

**2.1.1 Fields:**

* **user\_data**: Stores the user's financial data, including income, expenses, and savings goals.
* **report\_content**: Contains the content of the generated financial report, including budget allocation recommendations and progress towards savings goals.

### **2.1.2 Methods:**

**\_init\_(self, income)**:

**Purpose:** Initializes a SavingsRecommendation object with the user's specified income.

**Parameters:**

**income**: The user's monthly income.

**Behavior:** Constructs an instance of the SavingsRecommendation class with the provided income value.

**percentage\_money\_for\_taxes(self, tax\_rates\_percentage)**:

**Purpose:** Computes the amount of money allocated for taxes based on the specified tax rate percentage.

**Parameters:**

**tax\_rates\_percentage**: The percentage of income allocated for taxes.

**Returns:**

The calculated amount of money allocated for taxes.

**percentage\_money\_for\_needs(self, needs\_percentage)**:

**Purpose:** Calculates the amount of money allocated for needs based on the specified percentage.

**Parameters:**

**needs\_percentage**: The percentage of income allocated for needs.

**Returns:**

The calculated amount of money allocated for needs.

**percentage\_money\_for\_wants(self, money\_percentage\_for\_wants)**:

**Purpose:** Determines the amount of money allocated for wants based on the specified percentage.

**Parameters:**

**money\_percentage\_for\_wants**: The percentage of income allocated for wants.

**Returns:**

The calculated amount of money allocated for wants.

**2.1.3 Function:**

#### **main()**

* **Purpose:** The main function of the program, orchestrating user interactions and program flow.
* **Parameters:** None
* **Returns:** None
* **Behavior:**

Prompts the user to input their monthly salary.

Creates an instance of the ‘**SavingsRecommendation**’ class with the provided income.

Presents a menu to the user, allowing them to choose between calculating the budget allocation or exiting the program.

If the user chooses to calculate the budget allocation:

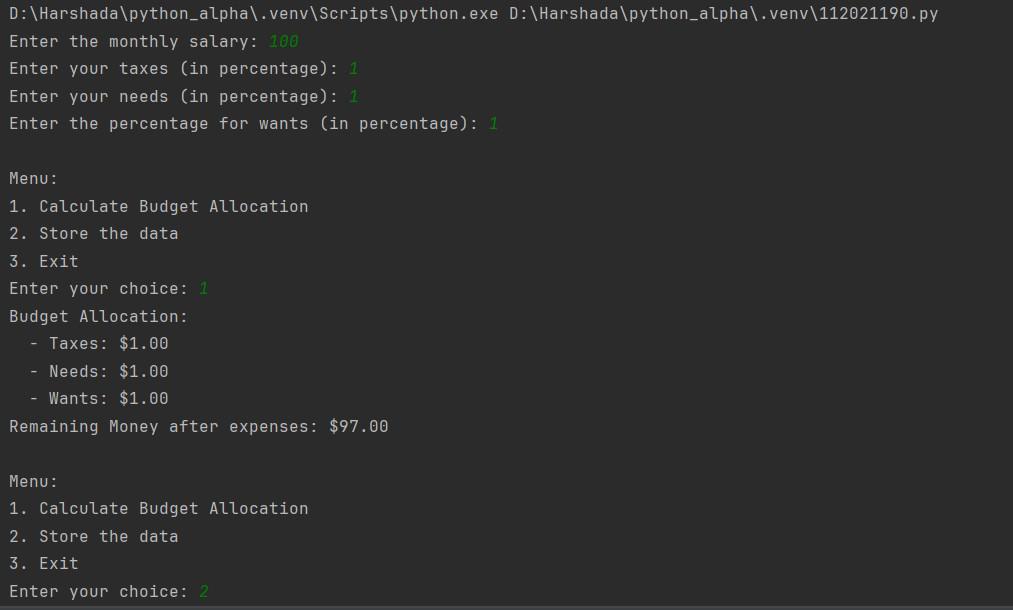
* + - Calls the ‘c**alculate\_budget\_allocation**’ method of the **‘SavingsRecommendation**’ instance.
    - Displays the result to the user.

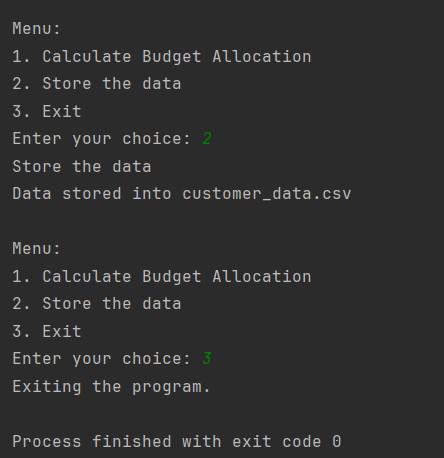
If the user chooses to exit the program:

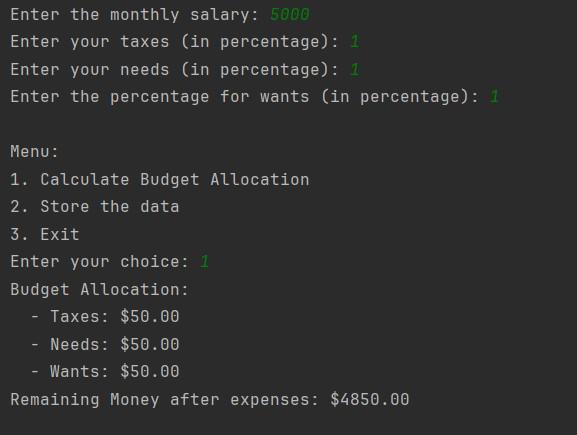
* + - Prints a message indicating program termination.
    - Exits the program.

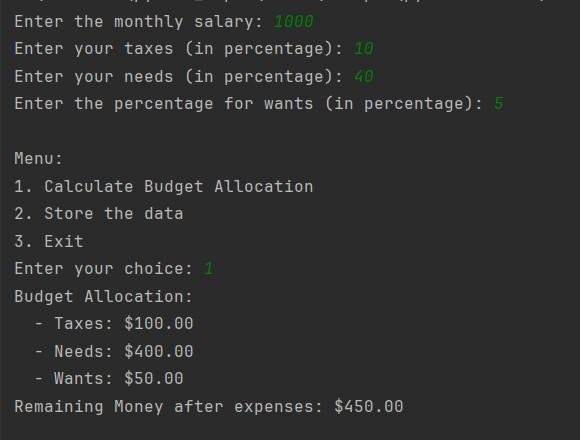
**Chapter 3 - Results**

**3.1 Result**.









**Chapter 4 Conclusions**

The ‘**SavingsRecommendation’** class, along with the ‘**main()’** function, provides a user-friendly tool for financial planning. It promotes responsible spending habits by suggesting allocations for essential expenses, discretionary spending, and savings. Users can utilize this program to gain insights into optimizing their budget and achieving their financial goals effectively