A C++ Project Report on

SAVING GOAL TRACKER WITH MONTHLY

CONTRIBUTIONS



BSCY SAVING GOAL TRACKER WITH MONTHLY

CONTRIBUTIONS

by

JUNAID AHMED BCY243053

WAQAS UR REHMAN BCY 243042

A Project Report submitted to the

DEPARTMENT OF ELECTRICAL AND COMPUTER

ENGINEERING

in partial fulfillment of the requirements for the degree of

BACHELORS OF SCIENCE IN CYBER SECURITY

Faculty of Engineering

Capital University of Science & Technology,

Islamabad

January, 2025

SAVING GOAL TRACKER WITH MONTHLY

CONTRIBUTIONS



by

JUNAID AHMED BCY243053

WAQAS UR REHMAN BCY 243042

A Project Report submitted to the

DEPARTMENT OF ELECTRICAL AND COMPUTER

ENGINEERING

in partial fulfillment of the requirements for the degree of

BACHELORS OF SCIENCE IN CYBER SECURITY

Faculty of Engineering

Capital University of Science & Technology,

Islamabad

January, 2025

##### Copyright © 2022 by CUST Student

All rights reserved. Reproduction in whole or in part in any form requires the prior written permission of JUNAID AHMED BCY243053, WAQAS UR REHMAN BCY243042 or designated representative.

**DECLARATION**

It is declared that this is an original piece of our own work, except where otherwise acknowledged in text and references. This work has not been submitted in any form for another degree or diploma at any university or other institution for tertiary education and shall not be submitted by us in future for obtaining any degree from this or any other University or Institution.

JUNAID AHMED

Reg. No. BCY243053

WAQAS UR REHMAN

Reg. No. BCY2430042

January, 2025

**CERTIFICATE OF APPROVAL**

It is certified that the project titled “railway track crack detection sytem based on image processing” carried out by JUNAID AHMED BCY243053, WAQAS UR REHMAN BCY243053 under the supervision of Mr. Waqas Ayub Shah, Capital University of Science & Technology, Islamabad, is fully adequate, in scope and in quality, as a final year project for the degree of BS Electrical and Computer Engineering.

|  |  |
| --- | --- |
| Supervisor: | --------------------------------------  Mr. Waqas Ayub Shah  Lecturer  Department of Electrical and Computer Engineering  Faculty of Engineering  Capital University of Science & Technology, Islamabad |
| HoD: | -------------------------------------- |

##### Dr. Noor Mohammad Khan

##### Professor

Department of Electrical and Computer Engineering

##### Faculty of Engineering

##### Capital University of Science & Technology, Islamabad

**ACKNOWLEDGMENT**

We would like to thank “Mr. Waqas Ayub Shah” whose guidance helped to complete our report in the given time. Last but not the least, this project cannot be completed without the effort and co-operation of group members.

**ABSTRACT**

This program is a \*Savings Goal Tracker\* designed to assist users in setting and tracking their progress toward a financial savings goal. The user begins by inputting their desired \*savings goal\* and the \*number of months\* they plan to achieve it. Based on these inputs, the program calculates the required \*monthly contribution\* to meet the goal. The program then presents an interactive menu offering several options to manage the savings plan.

The menu provides the following options:

1. \*Track Progress\*: This option displays the savings accumulated each month, based on the monthly contribution. The program checks whether the savings goal has been met by the end of the specified time period, and provides feedback accordingly.

2. \*Update Savings Goal\*: The user can modify the savings goal. Once updated, the program recalculates the required monthly contribution based on the new goal.

3. \*Update Number of Months\*: The user can change the number of months they plan to save, and the program recalculates the monthly contribution accordingly.

4. \*Exit\*: This option allows the user to exit the program.

The program includes \*error handling\* to ensure that invalid input is avoided. It checks for non-positive values for the savings goal and the number of months, prompting the user to enter valid inputs. If the user tries to track progress without specifying a valid number of months, the program will ask for corrections.

Additionally, the program \*restarts the savings tracking process\* each time the user interacts with the menu, allowing them to modify their plans without restarting the program. When the user chooses the \*Track Progress\* option, it simulates the accumulation of savings over the months and provides monthly updates. After all months are processed, the program checks if the savings goal has been met and notifies the user of the result. If the goal is achieved, a congratulatory message is displayed; otherwise, the program informs the user that the goal has not been reached.

Overall, this \*Savings Goal Tracker\* is a simple and effective tool for individuals looking to plan and track their savings over time. It provides flexibility, allowing users to adjust their goals and timeframes as needed, and helps maintain motivation by showing progress toward achieving financial objectives. The program continues running in a loop until the user decides to exit, offering ongoing support for savings management.

TContents

[Chapter 1: 5](#_Toc187866604)

[Introduction 5](#_Toc187866605)

[1.1 Overview 5](#_Toc187866606)

[1.2 Project Idea 5](#_Toc187866607)

[1.3 Purpose of Project 5](#_Toc187866608)

[1.4 Project Significance 5](#_Toc187866609)

[1.5 Application of the Project 5](#_Toc187866610)

[1.6 Project Plan 6](#_Toc187866611)

[1.6.1 Define the Project Requirements 6](#_Toc187866612)

[1.6.2 Design the Program Structure 6](#_Toc187866613)

[1.6.3 Implement the Program 6](#_Toc187866614)

[1.6.4 Test the Program 6](#_Toc187866615)

[1.6.5 Debug the Program 7](#_Toc187866616)

[1.6.6 Document the Program 7](#_Toc187866617)

[Chapter 2: 7](#_Toc187866618)

[2.1 Program Features 7](#_Toc187866619)

[2.2 Program Structure 8](#_Toc187866620)

[2.3 Program Advantages 9](#_Toc187866621)

[2.4 Program Limitations 10](#_Toc187866622)

[Chapter 3: 10](#_Toc187866623)

[3.1 Overview of Program Structure 11](#_Toc187866624)

[3.2 High-Level Flow of the Program 11](#_Toc187866625)

[3.4 Program Design Considerations 12](#_Toc187866626)

[3.5 Example Flow 12](#_Toc187866627)

[3.6 Additional Points to Add 13](#_Toc187866628)

[Chapter 04: 14](#_Toc187866629)

[Project code 14](#_Toc187866630)

[Chapter 05: 16](#_Toc187866631)

[Chapter 6: 17](#_Toc187866632)

[Conclusion 17](#_Toc187866633)

**Chapter 1:**

## Introduction

## 1.1 Overview

The Savings Goal Tracker is a simple C++ program designed to help users set, monitor, and adjust their savings goals over a specified period. By allowing users to define their financial goals and track progress month by month, the program provides an interactive tool to plan and evaluate their savings journey. The program features options to update the savings goal, modify the time frame for reaching the goal, and track the savings progress as time progresses.

## 1.2 Project Idea

The idea behind this project is to create a straightforward application that helps users manage their savings by calculating the required monthly contribution and allowing them to monitor progress over time. The program is interactive, allowing the user to make real-time changes to their goal and the months to achieve it. The user can also track whether they have achieved their savings goal by the end of the defined period.

## 1.3 Purpose of Project

The purpose of the Savings Goal Tracker is to provide an easy-to-use tool that assists individuals in achieving their financial objectives. This program helps users break down large savings goals into manageable monthly contributions and track progress consistently. The tool also allows for flexibility in adjusting the goal or the number of months, ensuring users can adapt to their changing financial circumstances.

## 1.4 Project Significance

The significance of this project lies in its ability to offer practical assistance to individuals who are trying to achieve a financial target. By providing an intuitive interface to track savings progress and adjust savings plans as needed, the program serves as a helpful resource for personal financial planning. It encourages disciplined savings behavior and can be applied by anyone who wants to reach a specific financial goal, whether for an emergency fund, a vacation, or a larger investment.

## 1.5 Application of the Project

This project has several potential applications:

- \*Personal Finance Management\*: Individuals can use the program to set and track their personal savings goals, helping them stay organized and focused.

- \*Financial Planning for Students\*: Students can utilize the tracker to manage savings for tuition, books, or future expenses.

- \*General Savings Tracking\*: The program can be used by anyone looking to save for specific objectives, such as buying a car, a home, or building an emergency fund.

- \*Budgeting\*: It helps users understand how much they need to save each month to meet long-term financial targets, aiding in overall budgeting efforts.

## 1.6 Project Plan

### 1.6.1 Define the Project Requirements

The first step is to define the requirements of the program. The user should be able to:

- Input their total savings goal and the number of months they want to save.

- Receive the monthly contribution amount required to meet the goal.

- Track progress on a month-by-month basis.

- Update their savings goal or number of months and adjust the contribution amount accordingly.

- Handle invalid inputs (e.g., negative values for goal or months).

- Exit the program gracefully.

### 1.6.2 Design the Program Structure

The program structure is designed around user interaction. A \*main menu\* offers multiple options, including tracking progress, updating the goal, modifying the time frame, and exiting the program. The program will also need variables to store the savings goal, monthly contribution, current savings, and number of months.

The structure will follow a clear flow:

- Initialization of user inputs.

- Calculation of monthly contribution.

- Looped menu-driven interface for tracking, updating, and exiting.

### 1.6.3 Implement the Program

The program is implemented in C++ using basic concepts such as \*variables, \*\*loops, \*\*conditional statements, and \*\*input/output\*. The code follows a procedural approach with functions that handle user input, calculations, and displaying the progress. Each section of the menu triggers a different action, such as updating values, recalculating contributions, or displaying the savings progress.

### 1.6.4 Test the Program

Once the program is implemented, it will undergo testing to ensure that all functionalities work as expected. The program will be tested for:

- \*Correct monthly contribution calculation\* based on the user’s inputs.

- \*Accurate progress tracking\*, including the correct accumulation of savings.

- \*Correct handling of invalid inputs\*, ensuring the program responds appropriately when invalid data is entered.

### 1.6.5 Debug the Program

Any issues discovered during testing will be addressed through debugging. Common bugs might include incorrect calculations, input validation errors, or issues with the program flow (e.g., skipping menu options). Each bug will be fixed by analyzing the code, adjusting logic or variable handling, and rerunning the program to ensure correctness.

### 1.6.6 Document the Program

Documentation will be created for the program, outlining:

- \*Code comments\* explaining the functionality of different sections.

- A \*user manual\* detailing how to use the program.

- \*Error handling procedures\*, outlining how the program handles invalid input and guides the user through corrections.

- \*Test cases\* that show different scenarios, such as entering invalid months or goals, and how the program behaves in these cases.

This documentation will ensure the program is understandable and user-friendly for anyone who wishes to use or modify it in the future.

Chapter 2:

Program Objectives

## 2.1 Program Features

The \*Savings Goal Tracker\* program includes several key features designed to help users set, track, and adjust their savings goals. These features are:

1. \*User Input for Savings Goal and Time Frame\*:

- The user can enter their desired \*savings goal\* (total amount they wish to save).

- The user specifies the \*number of months\* they intend to achieve the goal.

2. \*Monthly Contribution Calculation\*:

- The program automatically calculates the \*monthly contribution\* required to meet the savings goal, based on the total goal amount and the number of months.

3. \*Track Savings Progress\*:

- The program allows the user to track the \*accumulation of savings month by month\*.

- After the completion of each month, the program calculates and displays the updated savings, helping users visualize their progress.

4. \*Modify Savings Goal\*:

- The user can update their savings goal at any time, and the program will recalculate the monthly contribution required to reach the new goal.

5. \*Modify Time Frame\*:

- The user can change the number of months to achieve the goal, which will update the required monthly contribution based on the new timeframe.

6. \*User-Friendly Menu\*:

- A simple, interactive menu allows the user to choose between tracking progress, updating their savings goal or months, and exiting the program.

7. \*Error Handling\*:

- The program checks for invalid input such as non-positive values for the savings goal or months and prompts the user to enter valid data.

8. \*Exit Option\*:

- The user can choose to exit the program at any time, ensuring they can leave the application gracefully.

## 2.2 Program Structure

The structure of the \*Savings Goal Tracker\* program follows a straightforward, modular approach:

1. \*Initialization\*:

- The program starts by prompting the user to input their \*savings goal\* and the \*number of months\* to achieve it.

- The monthly contribution is calculated by dividing the savings goal by the number of months.

**2. \*Main Menu Loop\*:**

- The program enters a loop that continuously displays a menu until the user decides to exit.

- The menu options include:

- \*Track progress\* (displays the savings accumulation over the months).

- \*Update savings goal\* (modifies the goal and recalculates the monthly contribution).

- \*Update number of months\* (modifies the time frame and recalculates the monthly contribution).

- \*Exit\* (ends the program).

3**. \*Functions\*:**

- Different sections of the program are broken into functions for better organization and readability:

- \*Input Validation\*: Ensures valid data is entered for the savings goal and number of months.

**- \*Savings Calculation**\*: Recalculates monthly contributions based on changes to the goal or time frame.

**- \*Tracking Progress**\*: Keeps track of the accumulated savings over time.

**4. \*User Interaction\*:**

- The user interacts with the program through console input and output, responding to prompts and choosing menu options.

**5. \*Exit Handling\*:**

- When the user selects the "Exit" option, the program terminates gracefully, ensuring no errors or incomplete states.

## 2.3 Program Advantages

The \*Savings Goal Tracker\* offers several advantages for users:

**1. \*Ease of Use\*:**

- The program is user-friendly, with a clear menu interface that guides the user through various options. The input and output are simple and intuitive, making it accessible even for individuals with limited technical knowledge.

**2. \*Real-Time Feedback\*:**

- Users receive immediate feedback on their savings progress, helping them stay motivated to meet their financial goals. The program shows monthly updates, so users can track their success over time.

**3. \*Flexible Adjustments\*:**

- The program provides flexibility by allowing users to update both their savings goal and the number of months to reach it. This makes the program adaptable to changes in the user’s financial circumstances or goals.

**4. \*Promotes Financial Planning\*:**

- By breaking down a large savings goal into monthly contributions, the program encourages better financial planning and budgeting. It helps users understand how much they need to save each month to achieve their goal.

**5. \*Error Handling\*:**

- The program ensures that invalid inputs (such as negative values for goal or months) are prevented, prompting users to correct the input. This prevents runtime errors and ensures the program operates smoothly.

**6. \*Affordable and Accessible\*:**

- Since it is implemented in C++, a widely available programming language, the program is accessible on almost any platform where C++ compilers are available, making it easy to use for a wide range of users.

## 2.4 Program Limitations

While the \*Savings Goal Tracker\* provides several advantages, it also has some limitations:

**1. \*Limited Functionality\*:**

- The program is relatively basic and lacks advanced features such as integration with real bank accounts, automated updates based on real interest rates, or detailed visual charts to display savings progress. It functions purely on user inputs and simple calculations.

* **2. \*No Persistence\*:**
* The program does not store data beyond the current session. If the user closes the program, all input data, including the savings goal, months, and progress, are lost. There is no file saving functionality or database support to retain user data.

**3. \*No Support for Variable Contributions\*:**

- The program assumes a fixed monthly contribution throughout the saving period. It does not account for fluctuations in the amount saved each month, such as extra contributions in some months or missed payments.

**4. \*User Dependency\*:**

- Since the program relies entirely on user input for updating and tracking savings, it may not be suitable for those who require more automated systems that track savings directly from their bank accounts or financial institutions.

**5. \*Limited Error Handling\*:**

- While basic error handling for invalid inputs is implemented, there could be more sophisticated handling, such as catching unexpected user errors or edge cases, like entering non-numeric characters or extremely large values.

**6. \*No Advanced Goal Tracking\*:**

- The program does not support more advanced tracking options, such as different types of savings goals (e.g., emergency fund vs. vacation fund) or categorizing savings progress. It treats all goals as a single, generic financial target.

These limitations suggest that while the \*Savings Goal Tracker\* is useful for simple savings planning, it could be expanded to include more features, error handling, and data persistence for future use.

Chapter 3:

Program Structure

## 3.1 Overview of Program Structure

The \*Savings Goal Tracker\* program is designed with simplicity and efficiency in mind. Its structure is broken down into clear sections to allow for easy understanding, modification, and maintenance. The program utilizes standard input/output operations, conditional statements, loops, and basic arithmetic operations to calculate and track the savings progress.

## 3.2 High-Level Flow of the Program

The flow of the program is designed to be interactive and straightforward. The user interacts with the program through a text-based menu system, where they can choose various options to input data, update information, track progress, and ultimately reach their savings goal. The program executes in a continuous loop until the user decides to exit. Here's an overview of the program flow:

Initialization Phase:

The user is prompted to enter their savings goal and number of months they want to save.

The monthly contribution required to reach the goal is calculated using the formula:

monthly contribution =savings goal months

monthly contribution= months savings goal

​

**3.3 Components of the Program**

**1. \*Variables\*:**

**- \*savings Goal\*:** Stores the total savings target.

**- \*monthly Contribution\*:** Holds the calculated monthly savings required to reach the goal.

**- \*current Savings\*:** Keeps track of the accumulated savings over time.

**- \*months\*:** The number of months allocated to achieve the savings goal.

**- \*choice\*:** Stores the user’s menu choice to determine which operation to perform.

2. \*Functions\*:

**- \*Input Handling\*:** The program handles user input using standard input/output functions (cin cout).

**- \*Calculation Functions:** Functions perform calculations to update the \*\*monthly contribution\* whenever the user updates the savings goal or the number of months.

**- \*Menu Handling\*:** A loop allows the program to continuously show the menu options and handle the user’s choices.

**- \*Progress Tracking\*:** After selecting the "Track Progress" option, a loop runs for the number of months, updating and displaying the savings progress.

## 3.4 Program Design Considerations

**1. \*Modularity\*:**

- The program is designed to be modular, meaning that different operations (e.g., tracking progress, updating the goal, updating the months) are handled separately to keep the code organized and readable.

**2. \*User Interaction\*:**

- The program is built to be interactive, allowing the user to make changes as needed. If invalid inputs are entered, the program prompts the user for valid data, ensuring smooth execution.

**3. \*Error Handling\*:**

- Basic error handling is incorporated to catch invalid inputs, such as negative or zero values for the savings goal or months. This prevents the program from encountering runtime errors or incorrect results.

**4. \*Flexibility\*:**

- Users can modify both the savings goal and the number of months at any point, which makes the program adaptable to changes in their financial circumstances.

**5. \*Efficiency\*:**

- The program is designed to be lightweight and efficient, calculating the monthly contribution in constant time and tracking progress in a simple loop.

## 3.5 Example Flow

Here is an example of how the program would flow:

**1. \*User Input\*:**

Enter your savings goal: $5000

Enter the number of months to reach your goal: 12

Monthly contribution needed to reach your goal: $416.67

**2. \*Menu Options\*:**

Menu

-----

1. Track progress

2. Update savings goal

3. Update number of months

4. Exit

Enter your choice: 1

**3. \*Tracking Progress\*:**

Month 1: Current savings = $416.67

Month 2: Current savings = $833.34

Month 3: Current savings = $1250.01

...

Month 12: Current savings = $5000.00

Congratulations! You have reached your savings goal.

**4. \*Update Savings Goal\*:**

Enter new savings goal: $6000

Monthly contribution updated to: $500.00

**5. \*Exit\*:**

Exiting program.

## 3.6 Additional Points to Add

**1. \*Data Persistence\*:**

- One area for future enhancement is adding \*data persistence\*. Currently, the program does not save data between sessions. Implementing functionality to save user data (e.g., savings goal, months, and progress) to a file would allow users to resume their progress after closing the program.

**2. \*Visual Feedback\*:**

- Incorporating simple graphical or textual representations of progress, such as a bar chart or a percentage complete, could enhance user experience by providing visual feedback.

**3. \*Interest Rate and Variable Contributions\*:**

- A possible extension would be to allow users to account for \*interest rates\* or \*variable monthly contributions\* (e.g., saving more money in some months and less in others). This would provide a more realistic simulation of savings.

**4. \*Security Features\*:**

- If the program were expanded to handle real financial data, it would need additional \*security features\* to ensure that user information is kept secure.

**5. \*User Profiles\*:**

- Adding the ability for multiple \*user profiles\* could make the program more flexible for families or individuals who want to track different savings goals simultaneously.

# Chapter 04:

## Project code

#include <iostream>

using namespace std;

int main()

{

double savingsGoal, monthlyContribution, currentSavings = 0;

int months, choice;

cout << "Savings Goal Tracker" << endl;

cout << "---------------------" << endl;

cout << "Enter your savings goal: $";

cin >> savingsGoal;

cout << "Enter the number of months to reach your goal: ";

cin >> months;

monthlyContribution = savingsGoal / months;

cout << "Monthly contribution needed to reach your goal: $" << monthlyContribution << endl;

while (true)

{

cout << "\nMenu" << endl;

cout << "-----" << endl;

cout << "1. Track progress" << endl;

cout << "2. Update savings goal" << endl;

cout << "3. Update number of months" << endl;

cout << "4. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

if (choice == 1)

{

if (months <= 0)

{

cout << "Invalid number of months. Please update." << endl;

} else

{

for (int i = 1; i <= months; i++)

{

currentSavings += monthlyContribution;

cout << "Month " << i << ": Current savings = $" << currentSavings << endl;

}

if (currentSavings >= savingsGoal)

{

cout << "Congratulations! You have reached your savings goal." << endl;

}

else

{

cout << "Sorry, you have not reached your savings goal." << endl;

}

}

}

else if (choice == 2)

{

if (savingsGoal <= 0)

{

cout << "Invalid savings goal. Please enter a positive value." << endl;

} else

{

cout << "Enter new savings goal: $";

cin >> savingsGoal;

monthlyContribution = savingsGoal / months;

cout << "Monthly contribution updated to: $" << monthlyContribution << endl;

}

}

else if (choice == 3)

{

if (months <= 0)

{

cout << "Invalid number of months. Please enter a positive value." << endl;

}

else

{

cout << "Enter new number of months: ";

cin >> months;

monthlyContribution = savingsGoal / months;

cout << "Monthly contribution updated to: $" << monthlyContribution << endl;

}

}

else if (choice == 4)

{

return 0;

}

else

{

cout << "Invalid choice. Please try again." << endl;

}

}

return 0;

}

# 

# Chapter 05:

START

INPUT SAVING GOAL AMOUNT AND MONTHLY CONTRIBUTION

INITIALIZE TRACKER

ADD MONTHLY CONTRIBUTION

DISPLAY PROGRESS

IF

GOAL REACHED

F T

ASK TO ADD ANOTHER MONTH’S CONTRIBUTION

GOAL REACHED

END

# Chapter 6:

## Conclusion

The \*Savings Goal Tracker\* program is a straightforward and practical tool designed to assist individuals in managing their financial goals. By enabling users to input their savings targets, calculate monthly contributions, track their savings progress, and make adjustments to their goals and timeframes, the program serves as an essential resource for those aiming to develop disciplined savings habits.

**Key Accomplishments**

**1. \*Simplicity and Usability\*:**

The program’s user-friendly interface ensures that individuals with little to no technical knowledge can easily navigate through the menu options and efficiently track their savings progress. By providing clear instructions and responding to user inputs in real-time, the program maintains a smooth and intuitive flow.

**2. \*Customization\*:**

The program is highly flexible, allowing users to update their savings goals or adjust the number of months to reach their target. This adaptability ensures that the program caters to the dynamic nature of personal finance, as users may encounter changes in their financial situation or timelines.

**3. \*Accurate Tracking and Calculation\*:**

The program offers accurate monthly savings calculations based on the user’s inputs, providing timely feedback on progress. By showing month-by-month accumulation, it offers a clear picture of how well the user is adhering to their savings plan and whether they are on track to meet their goal.

**4. \*Error Handling\*:**

Basic error handling ensures that the program prevents invalid inputs, prompting the user to enter valid data if necessary. This improves the overall user experience, ensuring that the program runs smoothly without any interruptions due to incorrect inputs.

**Limitations and Areas for Improvement**

While the program performs its intended functions effectively, there are several areas for potential enhancement:

**1. \*Data Persistence\*:**

The program currently lacks the ability to save user data between sessions. Introducing \*data persistence\*, such as saving progress to a file, would allow users to resume their savings plan from where they left off. This feature would add significant convenience for long-term use.

**2. \*Advanced Financial Features\*:**

The program could be enhanced by incorporating advanced financial features, such as \*interest rate calculations\* or the ability to make \*variable monthly contributions\*. This would make the program more suitable for users who want to simulate more complex savings scenarios, such as saving in an interest-bearing account or fluctuating their contributions each month.

**3. \*Visual Representation of Progress\*:**

Adding simple visual tools, such as a \*progress bar\* or graphical chart, could provide a clearer and more engaging representation of the user's savings over time. This would make the program more visually appealing and improve the user experience.

**4. \*Security and User Profiles\*:**

For future iterations, the addition of \*user profiles\* and \*security features\* would allow multiple users to track their savings goals securely and separately. If the program were expanded to handle sensitive financial data, strong security measures would be necessary to protect user information.

**Conclusion**

In conclusion, the \*Savings Goal Tracker\* program is an effective tool for personal finance management, offering simplicity, flexibility, and accurate tracking of savings progress. It enables users to take control of their financial goals by breaking down large targets into manageable monthly contributions and offering real-time feedback on their progress.

While the program successfully addresses its primary objectives, there is room for improvement in terms of advanced features, data persistence, and user experience. Future enhancements could make the program more versatile, robust, and valuable for a wider range of users. Nonetheless, this project demonstrates the potential of simple software tools to help individuals develop better financial habits and achieve their savings goals.