**NIMS INSTITUTE OF ENGINEERING & TECHNOLOGY**

JAIPUR, RAJASTHAN - 303121



A PYTHON MINI-PROJECT REPORT

ON

**“COMPOUND INTEREST GUI CALCULATOR”**

Submitted by

NAME : UNGARLA SUJITH ROSHAN

REGISTER NO : 137993

**School of Computing & Artificial Intelligence**

**NIMS UNIVERSITY**

2023-2024

**NIMS INSTITUTE OF ENGINEERING & TECHNOLOGY**

JAIPUR, RAJASTHAN - 303121

**School of Computing & Artificial Intelligence**

****

CERTIFICATE

Certified that the PYTHON mini-project work entitled

**“COMPOUND INTEREST GUI CALCULATOR”**

is a bonafide workcarried out by

**UNGARLA SUJITH ROSHAN**

**137993**

The report has been approved as it satisfies the academic requirements in respect of PYTHON mini-project work prescribed for this course.

……………...…………………………

**Dr. Sasikala Dhamodaran / Dr. Pramod Kumar Bhatt / Mr. Pradeep Upadhyay**

PYTHON Mini-Project Coordinator

INDEX

**1.LIST OF ABBREVIATIONS ……..01**

**2.ABSTRACT ……..02**

**3.INTRODUCTION ………03**

**4.SCOPE ………04**

**5.IMPORTANCE ………...05**

**6.PROS AND CONS …………06-07**

**7.IMPLEMENTATION ……….08**

**8.RESULT & ANALYSIS ……….09-10**

**9.CONCLUSION ……….11**

**10.FUTURE ENCHANCEMENT**

**11.REFERENCE**

**LIST OF ABBREVIAITIONS**

**P: Principal**  
The initial amount of money invested or borrowed.

**r: Rate**  
The annual interest rate (expressed as a decimal). For example, 5% would be 0.05.

**n: Number of Compounding Periods**  
The number of times interest is applied to the principal within one year (e.g., annually, semi-annually, quarterly, monthly).

**t: Time**  
The duration for which the money is invested or borrowed, expressed in years.

**A: Amount**  
The total amount of money accumulated after interest is applied, including the principal and interest.

**CI: Compound Interest**  
The interest calculated on the initial principal, which also includes all accumulated interest from previous periods.

**Additional Terms:**

**Annual:** Pertaining to a one-year period; commonly used when discussing interest rates.

**Compounding:** The process of earning interest on both the original principal and the interest that has been added to it.

**Investment:** The action of putting money into financial schemes, shares, property, or a commercial venture with the expectation of achieving a profit.

**ABSTRACT**

This mini project presents a compound interest calculator implemented in Python, utilizing the Thinker library to facilitate user interactions. The program enables users to input essential financial parameters, including the principal amount, annual interest rate, number of compounding periods per year, and the investment duration in years.

The calculation is based on the compound interest formula, which determines the total amount accumulated after interest is applied. The user-friendly interface ensures a seamless experience, allowing individuals to understand how their investments grow over time through compounding.

The project serves as a practical introduction to financial concepts and programming fundamentals, demonstrating how to gather user input, perform calculations, and display results. Additionally, it lays the groundwork for further enhancements, such as incorporating error handling, saving results, or creating a graphical user interface development in Python programming.

**INTRODUCTION**

In today's financial landscape, understanding how investments grow over time is crucial for making informed decisions. One of the most powerful concepts in finance is compound interest, which allows money to grow exponentially over time as interest is earned not just on the principal amount but also on the accumulated interest from previous periods.

This project focuses on creating a compound interest calculator using Python. The calculator provides an intuitive way for users to input key variables such as the principal amount, annual interest rate, number of compounding periods per year, and the duration of the investment in years. By leveraging the formula for compound interest, the program calculates the total amount accrued after the specified period.

**The implementation of this calculator serves multiple purposes:**

**Educational Tool:** It helps users grasp the concept of compound interest and its impact on investments.

**Practical Application:** Users can simulate different investment scenarios, enhancing their understanding of financial growth over time.

**Scope**

The scope of this compound interest calculator project encompasses several key areas, providing a solid foundation for both financial education and programming practice. Here are the primary aspects of its scope:

**1. Financial Education:**

**Understanding Compound Interest:** The project aims to demystify the concept of compound interest, illustrating how it differs from simple interest and highlighting its significance in investment growth.

**Investment Scenarios:** Users can explore various investment scenarios by adjusting parameters such as principal, interest rate, compounding frequency, and time, helping them make informed decisions about their finances.

**2. Programming Skills Development:**

**Basic Python Proficiency:** Users will practice fundamental programming concepts such as functions, user input handling, and basic mathematical operations.

**Code Organization:** The project encourages structured programming by separating functionalities into distinct functions, promoting best practices in coding.

**3. User Interface and Experience:**

**Console-Based Interaction:** The initial implementation uses a simple command-line interface, making it accessible and easy to understand for beginners.

**User-Friendly Design:** By focusing on clarity and simplicity, the calculator ensures that users can easily navigate the inputs and understand the output.

**4. Future Enhancements:**

**Error Handling:** There is scope to implement input validation and error handling to enhance the robustness of the application.

### **IMPORTANCE**

The compound interest calculator project holds significant importance across several domains, including finance, education, and programming. Here are the key reasons highlighting its importance:

**Financial Literacy:**

Understanding Growth: The calculator helps individuals grasp how compound interest can lead to substantial growth over time, fostering better financial decision-making.

**Investment Planning:** By simulating different scenarios, users can understand the potential outcomes of their investments, aiding in long-term financial planning and wealth accumulation.

**Educational Value:**

**Conceptual Clarity:** The project serves as a practical tool for teaching the principles of compound interest, making abstract financial concepts more tangible and relatable.

**Engagement:** Interactive tools enhance learning by encouraging users to experiment with various inputs, reinforcing their understanding of how different factors affect investment returns.

**Programming Skills Development:**

**Hands-On Experience:** For beginners, this project provides a real-world application of programming concepts, making it easier to grasp fundamental skills in Python.

**Problem-Solving:** Developing the calculator promotes critical thinking and problem-solving skills as users work through the design and implementation of the application.

**Empowerment:** Understanding how compound interest works empowers individuals to take control of their financial futures, encouraging responsible saving and investment behaviors.

**Pros and Cons**

**Pros :**

**Enhanced Financial Literacy:**

Helps users understand the concept of compound interest, promoting better financial decision-making.

**User-Friendly:**

Simple and intuitive interface allows users to easily input data and receive instant results.

**Practical Application:**

Users can simulate various investment scenarios, aiding in effective financial planning.

**Educational Tool:**

Serves as a valuable resource for teaching fundamental financial principles in educational settings.

**Skill Development:**

Offers hands-on programming experience, enhancing coding skills and problem-solving abilities for beginners.

Users can input their own parameters, allowing for personalized financial calculations tailored to individual needs.

**Cons :**

**Limited Functionality:**

The basic calculator may not cover more complex financial scenarios, such as varying interest rates, fees, or taxes.

**Lack of Visual Representation:**

A console-based interface may not provide graphical representations (e.g., charts or graphs) to visualize growth over time.

No Error Handling:

Basic implementations may not include input validation, leading to potential user errors or **crashes.**

Difficulty in Adding Features:

Although you can expand the project (e.g., by adding graphs, saving results to files, etc.), the basic nature of the compound interest calculator may make it difficult to add features that feel truly meaningful. Adding extra features like amortization charts, payment schedules, or additional investment options can significantly complicate the project and potentially divert focus from the core learning experience.

Possible Over-Simplicity:

Depending on the audience or context in which you're working, a simple compound interest calculator might feel too trivial or underwhelming, especially if you're working on a portfolio of projects. There’s not a lot of depth or complexity involved that would make it stand out among other Python projects.

**IMPLEMENTATION**

**Code:**

import tkinter as tk

from tkinter import messagebox

def calculate\_compound\_interest():

    try:

        principal = float(principal\_entry.get())

        rate = float(rate\_entry.get())

        times\_compounded = int(times\_entry.get())

        years = int(years\_entry.get())

        amount = principal \* (1 + (rate / (100 \* times\_compounded))) \*\* (times\_compounded \* years)

        interest = amount - principal

        messagebox.showinfo("Result", f"Total Amount: ${amount:.2f}\nCompound Interest: ${interest:.2f}")

    except ValueError:

        messagebox.showerror("Input Error", "Please enter valid numerical values.")

root = tk.Tk()

root.title("Compound Interest Calculator")

tk.Label(root, text="Principal Amount ($):").grid(row=0, column=0, padx=10, pady=10)

principal\_entry = tk.Entry(root)

principal\_entry.grid(row=0, column=1)

tk.Label(root, text="Annual Interest Rate (%):").grid(row=1, column=0, padx=10, pady=10)

rate\_entry = tk.Entry(root)

rate\_entry.grid(row=1, column=1)

tk.Label(root, text="Times Compounded per Year:").grid(row=2, column=0, padx=10, pady=10)

times\_entry = tk.Entry(root)

times\_entry.grid(row=2, column=1)

tk.Label(root, text="Number of Years:").grid(row=3, column=0, padx=10, pady=10)

years\_entry = tk.Entry(root)

years\_entry.grid(row=3, column=1)

calculate\_button = tk.Button(root, text="Calculate", command=calculate\_compound\_interest)

calculate\_button.grid(row=4, column=0, columnspan=2, pady=20)

root.mainloop()

**RESULT**

**Sample -1**

**Input:**

Enter the principal amount: 1000

Enter the annual interest rate (in decimal, e.g., 0.05 for 5%): 0.05

Enter the number of times interest applied per year: 4

Enter the number of years: 5

**Output:**

The amount after 5 years is: 1283.68

**Sample - 2**

**Input:**

Enter the principal amount: 2500

Enter the annual interest rate (in decimal, e.g., 0.05 for 5%): 0.06

Enter the number of times interest applied per year: 12

Enter the number of years: 10

**Output:**

The amount after 10 years is: 4495.36

**ANALYSIS**

**Analysis of the Project:**

1. **Complexity**:
   * **Time Complexity**: O(1) – The calculation of compound interest is constant time; it does not depend on the size of input.
   * **Space Complexity**: O(1) – The space required is constant as we are only storing a few variables for calculation.
2. **Extensions**:
   * **Graphical Representation**: You can extend the project by adding a chart that visually represents how the principal grows over time (using libraries like matplotlib).
   * **Additional Features**: Allow the user to choose different types of interest calculation (e.g., simple interest vs compound interest).
3. **Usability**:
   * The calculator is simple to use, and the input fields are clearly labeled. It provides instant results based on user input.
4. **Potential Improvements**:
   * Allow users to reset all fields after calculation.
   * Validate inputs more thoroughly (e.g., prevent negative numbers).

**CONCLUSION**

compound interest calculator successfully illustrates the fundamental principles of compound interest, showcasing its significant impact on investment growth over time. By enabling users to input various financial parameters—such as principal amount, interest rate, compounding frequency, and duration—the tool effectively demonstrates how these factors interact to influence the total amount accrued.

**Key takeaways from the results include:**

Exponential Growth: Users can see firsthand how compound interest leads to exponential growth, reinforcing the importance of starting investments early.

Impact of Compounding Frequency: Different compounding frequencies can dramatically alter the final outcome, emphasizing the need for careful consideration in investment choices.

Long-Term Planning: The calculator highlights the benefits of long-term investing, encouraging users to think about their financial futures.

**Educational Value:** The project serves as a valuable educational tool, enhancing financial literacy by making complex concepts more accessible.

To expand the functionality and usability of the compound interest calculator, several enhancements can be considered:

**FUTURE ENHANCEMENT**

**Error Handling and Input Validation:**

Implement robust error handling to manage invalid inputs gracefully, ensuring users are guided to enter correct values.

**Graphical User Interface (GUI):**

Develop a GUI using libraries such as Tkinter or PyQt to provide a more visually appealing and user-friendly experience, including charts to visualize growth over time.

**Advanced Features:**

Include additional financial calculations, such as loan amortization, savings projections, and retirement planning tools, to provide a more comprehensive financial resource.

**Data Persistence:**

Allow users to save their calculations and scenarios for future reference, possibly integrating a database or file storage system.

**Scenario Comparison:**

Implement functionality to compare multiple investment scenarios side by side, helping users make informed decisions based on different assumptions.

**Educational Resources:**

Provide links to articles, videos, or tutorials on compound interest and personal finance topics, enhancing the educational value of the tool.

**Mobile Compatibility:**

Adapt the calculator for mobile devices, allowing users to access it on the go, which can increase its accessibility and usability.

**REFERENCES**

Here are some useful references and resources that can help further understand compound interest, finance, and programming concepts related to this project:

**Financial Concepts:**

Investopedia. (n.d.). Compound Interest. Retrieved from Investopedia

Khan Academy. (n.d.). Compound Interest. Retrieved from [Khan Academy](https://www.khanacademy.org/math/algebra/x2f8bb11595b61c86b2b7b42b48d086c2a1/compound-interest/v/compound-interest)

**Python Programming:**

Python Software Foundation. (n.d.). Python Official Documentation. Retrieved from [Python.org](https://docs.python.org/3/)

W3Schools. (n.d.). Python Functions. Retrieved from W3Schools

**Financial Calculators:**

Bankrate. (n.d.). Compound Interest Calculator. Retrieved from Bankrate

**General Financial Literacy:**

National Endowment for Financial Education. (n.d.). Financial Literacy Resources. Retrieved from [NEFE](https://www.nefe.org/)

**Python Projects:**

Real Python. (n.d.). Build a Simple Command-Line Application with Python. Retrieved from Real Python