## 

## G H RAISONI COLLEGE OF ENGINEERING & MANAGEMENT, WAGHOLI, PUNE

**(**An Empowered Autonomous Institute, Affiliated to Savitribai Phule Pune University,

Approved by AICT. New Delhi and Recognized by Govt. of Maharashtra.

NAAC ACCREDIATED A+ Grade)

**Department of Information Technology**

**Subject: Python Programming [23UQCMDP2302]**

Mini Project Report On

# “Randomized Password Generator”

### Academic Year 2024-25 Semester-III

**G H RAISONI COLLEGE OF ENGINEERING & MANAGEMENT, WAGHOLI, PUNE**

**(**An Empowered Autonomous Institute, Affiliated to Savitribai Phule Pune University,

Approved by AICT. New Delhi and Recognized by Govt. of Maharashtra.

NAAC ACCREDIATED A+ Grade)

**Department of Information Technology Academic Year 2024-25**

### Semester-III

**Guided By:-**

**Mrs. Archana Dongardive**

**Submitted by:**

1. SITB05 Kaustubh Kumbhakarn
2. SITB06 Kaustubh Kulkarni
3. SITB03 Gajanan Joshi
4. SITB22 Nagesh Mahajan
5. SITB02 Gagan Kasarlawad



**CERTIFICATE**

This is to certify that **“**Border Gateway protocol**”** embodies the original work done by **during Mr. Kaustubh Kumbhakarn , Mr. Gajanan Joshi , Mr. Kaustubh Kulkarni , Mr. Nagesh Mahajan , Mr. Gagan Kasarlawad** this project submission as a partial fulfilment of the requirement for the Mini Project using Python Programming subjects of Second Year B. Tech. Degree, III Semester, of Pune University during the academic year 2024-2025.

**Date:**

**Place:** Pune

**(Mrs. Archana Dongardive (Dr. Poonam Gupta)**

**Project Guide Head of Department**

### ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Mrs. Archana Dongardive** under whose valuable guidance and light of knowledge, we could complete this project.

We take this opportunity to thank all the staff members of Department Of Information Technology Engineering for their help whenever required. Finally we express sincere thanks to all those who have helped us directly or indirectly in many ways in completion of this project work and I would like to extend my Deep appreciation to all my group members, without their support and Coordination we would not have been able to complete this Project.

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr.No | TITLE | Page No |
| 1. | Split expenses among friends and calculate how much each person owes. | 1 |

**Project Statement: Expense Splitter System**

This Python program simulates an expense splitting system for friends, allowing users to calculate how much each person owes after sharing expenses. The program utilizes object-oriented programming principles, including inheritance and polymorphism, to effectively manage the expense-sharing process.

Features of the Program:

1. **Expense Class**:
   * **Purpose**: Acts as the base class for storing details about each expense, including the total amount and a list of participants.
   * **Method**: add\_expense() allows users to input details of an expense.
2. **User Class**:
   * **Purpose**: Represents each user in the system, storing their name and balance (how much they owe or are owed).
   * **Method**: update\_balance() updates the user's balance based on expenses shared.
3. **ExpenseSplitter Class**:
   * **Purpose**: Inherits from the Expense class and extends functionality to handle multiple expenses and calculate individual shares.
   * **Features**:
     + **Add Expense**: Allows users to input new expenses and specify who participated.
     + **Calculate Shares**: Computes how much each participant owes based on the total expense and number of participants.
     + **Display Balances**: Lists each user's current balance after all expenses have been processed.
4. **Error Handling**:
   * **Input Validation**: Ensures that user inputs are valid (e.g., non-negative amounts) and handles invalid inputs gracefully by prompting the user to re-enter correct data.
5. **Main Function**:
   * **Purpose**: Manages the overall flow of the expense splitter simulation, allowing users to add expenses, view balances, or exit as needed.

CODE

def split\_expenses():

print("Welcome to the Expense Splitter!")

# Input total expenses

try:

total\_expenses = float(input("Enter the total expenses: $"))

if total\_expenses < 0:

raise ValueError("Expenses cannot be negative.")

except ValueError as e:

print(f"Invalid input: {e}")

return

# Input number of friends

try:

num\_friends = int(input("Enter the number of friends: "))

if num\_friends <= 0:

raise ValueError("Number of friends must be greater than zero.")

except ValueError as e:

print(f"Invalid input: {e}")

return

# Calculate amount owed per person

amount\_per\_person = total\_expenses / num\_friends

print(f"\nEach person owes: ${amount\_per\_person:.2f}")

# Optional: To track individual contributions

contributions = {}

for i in range(num\_friends):

name = input(f"Enter the name of friend {i + 1}: ")

try:

contribution = float(input(f"Enter how much {name} contributed: $"))

contributions[name] = contribution

except ValueError as e:

print(f"Invalid input for {name}: {e}")

return

# Calculate how much each person owes or is owed

print("\nSummary of contributions:")

for name, contribution in contributions.items():

balance = contribution - amount\_per\_person

if balance < 0:

print(f"{name} owes: ${-balance:.2f}")

elif balance > 0:

print(f"{name} is owed: ${balance:.2f}")

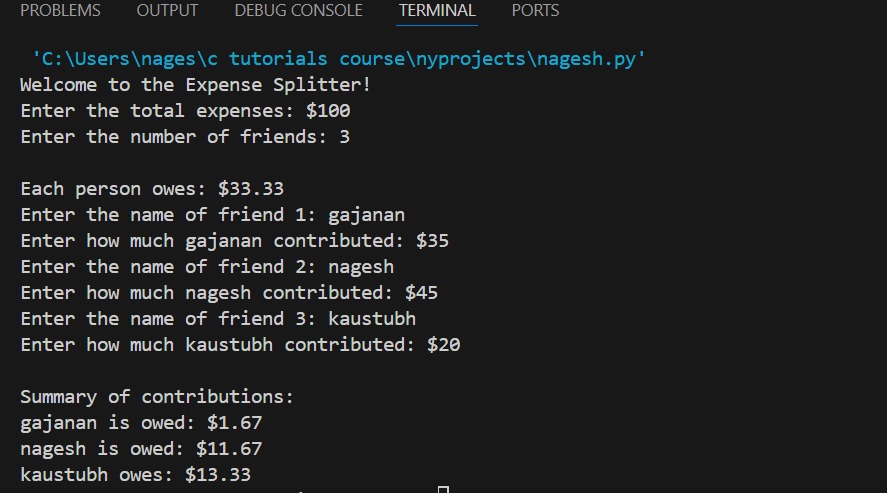
else:

print(f"{name} is settled up.")

if \_name\_ == "\_main\_":

split\_expenses()

OUTPUT



Thankyou