BAN 501

Module 1 Project B

Project Name	Advanced Marketing Campaign Budget Allocator
Project Due Date	Sunday by 11:59pm

Objectives

- Practice working with numeric variables, strings, and basic operators (+, -, *, /).
- Implement if statements and for loops to make decisions and repeat tasks.
- Create and utilize simple functions with parameters and return values.
- Perform basic calculations and manipulate data within lists.

Steps

1. Planning
    - Define a dictionary of available marketing channels (e.g., "Social Media", "Email Marketing", "Advertising") and assign a budget limit to each channel as an integer (e.g., “Social Media”: 5000).
    - Identify two factors influencing budget allocation:
        - target_audience (target audience options could include "Gen Z", "Millennials", "Gen X")
        - campaign_goal (campaign goal options could include "Brand Awareness", "Lead Generation", "Sales")
2. Coding
    - Start by introducing variables to store user input:
        - user_target_audience = input("Enter your target audience (Gen Z, Millennials, Gen X): ")
        - user_campaign_goal = input("Enter your campaign goal (Brand Awareness, Lead Generation, Sales): ")
    - Implement an if statement to branch based on the selected campaign_goal:
        - If Brand Awareness, allocate a 15% higher budget to "Social Media" and "Advertising".
        - If Lead Generation, allocate a 10% higher budget to "Email Marketing".
        - If Sales, allocate a higher a 5% budget to all channels.
    - Define a function called calculate_budget(channel, base_budget) that takes two arguments:
        - The desired marketing channel as a string (e.g., "Social Media").
        - The base budget for that channel as an integer (e.g., social_media_budget).
    - Inside the function, use conditional statements to adjust the base budget based on user_target_audience:
        - If Gen Z, increase the budget for "Social Media" slightly.
        - If Millennials, increase the budget for "Email Marketing" slightly.
        - If Gen X, increase the budget for "Advertising" slightly.
    - Use a for loop to iterate through the list of channels and call the calculate_budget function for each one, storing the returned values in a new list of allocated_budgets.

- Calculate the total remaining budget by subtracting the sum of allocated_budgets from the sum of all base budgets. (Allocated budgets are the budgets with the increased amounts. Base budgets are the original budget amounts.)

3. Testing and Refinement:
    - Test the program with different combinations of user_target_audience and user_campaign_goal.
    - Validate user input to ensure accurate data types and prevent errors.
    - Add comments to explain your code logic and improve readability.
4. Optional Enhancements:
    - Display the allocated budget for each channel and the remaining overall budget in a visually appealing format.
    - Implement a menu system to navigate different functionalities (e.g., budget allocation, budget overview).
    - Integrate real-world data like average costs for different marketing channels or historical campaign performance.

Additional Tips

- Use Python libraries like math for advanced calculations.
- Practice good coding practices like variable naming and code organization.
- Split complex tasks into smaller, more manageable functions.
- Don't hesitate to experiment and explore different approaches!

Grading Criteria

1. This project will be graded on an A*-F* scale (see the syllabus for details). To get high marks on this project, you must complete all requirements and have the project function perfectly.

Deliverable

Submit your project .py file on Canvas.