ChatGPT

Python Beginner Team Challenge: Grocery Store Checkout System

---

📊 Challenge Goal:

Simulate a real grocery store checkout using Python where a user selects multiple products, and the system checks if they can afford them.

Features to Implement:

1. **Product Catalog:**
2. Create a list of 20 grocery products with fixed prices.
3. Map each product to a number for easy selection.
4. **User Selection Menu:**
5. Display the product list with corresponding numbers.
6. Prompt the user to choose multiple products using numbers.
7. Allow the user to stop selecting products with a special input (e.g., 0).
8. **Total Cost Calculation:**
9. Store selected items in a list.
10. Calculate the total using a function.
11. **Budget Input & Comparison:**
12. Ask the user how much money they have.
13. Compare total cost to budget:
    - If within budget, show success message and change.
    - If not, show shortage and ask if they want to add more money.
    - If added amount is sufficient, proceed to payment.
14. **Final Output:**
15. Display selected items, total cost, amount paid, and change.
16. Show a thank-you message.

Tools & Concepts to Use:

- **Variables** for item prices and totals
- **Data Structures** like lists or dictionaries
- **Functions** inside a separate module (`budget.py`)
- **Loops** to allow multiple selections
- **Conditional Statements** for logic control
- **Input/Output** for interacting with the user
- **Namespaces and Modules** to keep code clean and reusable

Strategies for Team Execution:

1. **Divide Tasks:**
2. One person works on the product catalog and display.
3. Another writes the input loop for selection.
4. Another handles the calculation and comparison logic.
5. Someone integrates all parts and tests for errors.
6. **Write and Import Modules:**
7. Keep calculation functions in `budget.py`.
8. Import them using `from packages.budget import ...`.
9. **Test in Stages:**
10. First test product selection.
11. Then test budget input and total calculation.
12. Finally test the logic for budget comparison and change.
13. **Keep Code Simple:**
14. Use beginner-friendly syntax.
15. Comment your code for clarity.
16. **Discuss Before Coding:**
17. Review this document as a team.
18. Clarify each step together before anyone writes code.

Example Use Case (User Journey):

- User sees a list of 20 items.
- They choose item 1 (milk), item 2 (bread), item 4 (cereal).
- System adds the prices.
- User enters their available money.
- System compares and responds accordingly.
- Transaction completes with a final message.

Let's Build It Together!

Use this as your shared blueprint. Ask questions, test ideas, and help each other debug.

Happy Coding 💻!