




main.py



Run

```
1 import json
2
3 # Initialize an empty list to store
  transactions
4 transactions = []
5
6 # Function to record an income or expense entry
7 def record_transaction():
8     transaction_type = input("Enter transaction
      type (Income/Expense): ").lower()
9     category = input("Enter category: ")
10    amount = float(input("Enter amount: "))
11    date = input("Enter date (YYYY-MM-DD): ")
12
13    transaction = {
14        "type": transaction_type,
15        "category": category,
16        "amount": amount,
17        "date": date,
18    }
19
20    transactions.append(transaction)
21    save_transactions()
22
23 # Function to calculate the remaining budget
24 def calculate_budget():
25     income = sum(transaction["amount"] for
      transaction in transactions if
      transaction["type"] == "income")
26     expenses = sum(transaction["amount"] for
      transaction in transactions if
      transaction["type"] == "expense")
27     remaining_budget = income - expenses
28
29     print(f"Income: {income}")
30     print(f"Expenses: {expenses}")
31     print(f"Remaining Budget:
      {remaining_budget}")
32
33 # Function to display expense analysis
34 def expense_analysis():
35     expense_categories = {}
36     for transaction in transactions:
37         if transaction["type"] == "expense":
38             category = transaction["category"]
39             amount = transaction["amount"]
40             if category in expense_categories:
41                 expense_categories[category] +=
                  amount
42             else:
43                 expense_categories[category] =
                  amount
44
45     print("Expense Analysis:")
46     for category, amount in expense_categories
      .items():
47         print(f"{category}: {amount}")
48
49 # Function to save transactions to a JSON file
50 def save_transactions():
51     with open("transactions.json", "w") as file
      :
52         json.dump(transactions, file)
53
54 # Load transactions from the JSON file when the
  application starts
55 try:
56     with open("transactions.json", "r") as file
      :
57         transactions = json.load(file)
58 except FileNotFoundError:
59     transactions = []
60
```

Shell

Clear

Personal Budget Tracker

1. Record Transaction

2. Calculate Budget

3. Expense Analysis

4. Exit

Enter your choice: 1

Enter transaction type (Income/Expense): income

Enter category: salary

Enter amount: 3000

Enter date (YYYY-MM-DD): 2023-10-01

Personal Budget Tracker

1. Record Transaction

2. Calculate Budget

3. Expense Analysis

4. Exit

Enter your choice: 1

Enter transaction type (Income/Expense): expense

Enter category: groceries

Enter amount: 200

Enter date (YYYY-MM-DD): 2023-10-03

Personal Budget Tracker

1. Record Transaction

2. Calculate Budget

3. Expense Analysis

4. Exit

Enter your choice: 2

Income: 3000.0

Expenses: 200.0

Remaining Budget: 2800.0

Personal Budget Tracker

1. Record Transaction

2. Calculate Budget

3. Expense Analysis

4. Exit

Enter your choice: 3

Expense Analysis:

groceries: 200.0

Personal Budget Tracker

1. Record Transaction

2. Calculate Budget

3. Expense Analysis

4. Exit

Enter your choice: |



Clear

```
Personal Budget Tracker
1. Record Transaction
2. Calculate Budget
3. Expense Analysis
4. Exit
Enter your choice: 1
Enter transaction type (Income/Expense): income
Enter category: salary
Enter amount: 3000
Enter date (YYYY-MM-DD): 2023-10-01
Personal Budget Tracker
1. Record Transaction
2. Calculate Budget
3. Expense Analysis
4. Exit
Enter your choice: 1
Enter transaction type (Income/Expense): expense
Enter category: groceries
Enter amount: 200
Enter date (YYYY-MM-DD): 2023-10-03
Personal Budget Tracker
1. Record Transaction
2. Calculate Budget
3. Expense Analysis
4. Exit
Enter your choice: 2
Income: 3000.0
Expenses: 200.0
Remaining Budget: 2800.0
Personal Budget Tracker
1. Record Transaction
2. Calculate Budget
3. Expense Analysis
4. Exit
Enter your choice: 3
Expense Analysis:
groceries: 200.0
Personal Budget Tracker
1. Record Transaction
2. Calculate Budget
3. Expense Analysis
4. Exit
Enter your choice: 1
```