**Task 2**

**Description**: Develop a console-based budget tracker that allows users to manage their expenses and income.

**Features:**

1. Expense and Income Entry: Allow users to input expenses and income with categories and amounts.

2. Budget Calculation: Calculate the remaining budget after deducting expenses from income.

3. Expense Analysis: Provide insights by categorizing expenses and displaying spending trends.

4. Data Persistence: Store transactions in a file/database for tracking over time.

Creating a console-based budget tracker in Python with the specified features involves similar steps to the to-do list application. Here's a step-by-step guide to developing this application:

**Step-by-Step Implementation**

1. \*Setup Project Structure\*

Create a new directory for your project and set up the initial structure:

sh

mkdir budget\_tracker

cd budget\_tracker

touch budget.py transactions.json

2. \*Expense and Income Entry\*

Implement functions to allow users to input expenses and income with categories and amounts.

3. \*Budget Calculation\*

Implement functions to calculate the remaining budget after deducting expenses from income.

4. \*Expense Analysis\*

Provide insights by categorizing expenses and displaying spending trends.

5. \*Data Persistence\*

Use a JSON file to store transactions for persistence.

**Full Implementation in budget.py**

python

import json

import os

from datetime import datetime

TRANSACTIONS\_FILE = 'transactions.json'

# Load transactions from file

def load\_transactions():

if os.path.exists(TRANSACTIONS\_FILE):

with open(TRANSACTIONS\_FILE, 'r') as file:

return json.load(file)

return []

# Save transactions to file

def save\_transactions(transactions):

with open(TRANSACTIONS\_FILE, 'w') as file:

json.dump(transactions, file, indent=4)

# Add a new transaction

def add\_transaction(transactions, transaction\_type, category, amount):

transaction = {

'type': transaction\_type,

'category': category,

'amount': amount,

'date': datetime.now().strftime('%Y-%m-%d %H:%M:%S')

}

transactions.append(transaction)

save\_transactions(transactions)

# Calculate the remaining budget

def calculate\_budget(transactions):

income = sum(t['amount'] for t in transactions if t['type'] == 'income')

expenses = sum(t['amount'] for t in transactions if t['type'] == 'expense')

return income - expenses

# Analyze expenses by category

def analyze\_expenses(transactions):

categories = {}

for transaction in transactions:

if transaction['type'] == 'expense':

if transaction['category'] not in categories:

categories[transaction['category']] = 0

categories[transaction['category']] += transaction['amount']

for category, amount in categories.items():

print(f"Category: {category}, Amount: {amount}")

# Display transactions

def list\_transactions(transactions):

if not transactions:

print("No transactions available.")

return

for idx, transaction in enumerate(transactions):

print(f"{idx + 1}. [{transaction['type']}] {transaction['category']} - ${transaction['amount']} on {transaction['date']}")

**# Main command-line interface**

def main():

transactions = load\_transactions()

while True:

print("\nBudget Tracker Application")

print("1. Add income")

print("2. Add expense")

print("3. Calculate budget")

print("4. Analyze expenses")

print("5. List transactions")

print("6. Exit")

choice = input("Enter your choice: ")

if choice == '1':

category = input("Income category: ")

amount = float(input("Amount: "))

add\_transaction(transactions, 'income', category, amount)

elif choice == '2':

category = input("Expense category: ")

amount = float(input("Amount: "))

add\_transaction(transactions, 'expense', category, amount)

elif choice == '3':

remaining\_budget = calculate\_budget(transactions)

print(f"Remaining Budget: ${remaining\_budget}")

elif choice == '4':

analyze\_expenses(transactions)

elif choice == '5':

list\_transactions(transactions)

elif choice == '6':

break

else:

print("Invalid choice. Please try again.")

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Explanation of Features**

1. \*Expense and Income Entry\*: Functions to add income and expenses with categories and amounts (add\_transaction).

2. \*Budget Calculation\*: Function to calculate the remaining budget by deducting expenses from income (calculate\_budget).

3. \*Expense Analysis\*: Function to analyze and display expenses by category (analyze\_expenses).

4. \*Data Persistence\*: Transactions are saved to and loaded from a JSON file (load\_transactions, save\_transactions).

**Running the Application**

To run the application, simply execute the script:

sh

python budget.py

This setup provides a functional console-based budget tracker with the specified features, enabling users to manage their finances effectively.