PROJECT-3

\*\*\*\*\*\*\*\*\*\*

Expense Tracker

PROGRAM:

\*\*\*\*\*\*\*\*\*

# Expense Tracker Application

import datetime

class ExpenseTracker:

def \_\_init\_\_(self):

# Initialize an empty list to store expenses

self.expenses = []

def add\_expense(self, amount, category, description):

"""

Add a new expense to the tracker.

Args:

amount (float): The amount spent.

category (str): The category of the expense.

description (str): A brief description of the expense.

"""

expense = {

"date": datetime.date.today().strftime("%Y-%m-%d"),

"amount": amount,

"category": category,

"description": description,

}

self.expenses.append(expense)

print("\nExpense added successfully!")

def view\_expenses(self):

"""

Display all recorded expenses.

"""

if not self.expenses:

print("\nNo expenses recorded yet.")

return

print("\nRecorded Expenses:")

for idx, expense in enumerate(self.expenses, start=1):

print(

f"{idx}. Date: {expense['date']}, Amount: {expense['amount']}, "

f"Category: {expense['category']}, Description: {expense['description']}"

)

def analyze\_expenses(self):

"""

Analyze and summarize expenses by category and total amount.

"""

if not self.expenses:

print("\nNo expenses to analyze.")

return

category\_totals = {}

total\_expense = 0

for expense in self.expenses:

total\_expense += expense["amount"]

category\_totals[expense["category"]] = category\_totals.get(expense["category"], 0) + expense["amount"]

print("\nExpense Analysis:")

print(f"Total Expenses: {total\_expense:.2f}")

print("Category-wise Breakdown:")

for category, total in category\_totals.items():

print(f" {category}: {total:.2f}")

def menu(self):

"""

Display the menu and handle user input.

"""

while True:

print("\nExpense Tracker Menu")

print("1. Add Expense")

print("2. View Expenses")

print("3. Analyze Expenses")

print("4. Exit")

choice = input("Enter your choice: ").strip()

if choice == "1":

try:

amount = float(input("Enter amount spent: ").strip())

category = input("Enter category (e.g., Food, Transport): ").strip()

description = input("Enter a brief description: ").strip()

self.add\_expense(amount, category, description)

except ValueError:

print("Invalid input. Please enter valid data.")

elif choice == "2":

self.view\_expenses()

elif choice == "3":

self.analyze\_expenses()

elif choice == "4":

print("Thank you for using the Expense Tracker. Goodbye!")

break

else:

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

# Entry point of the program

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

tracker = ExpenseTracker()

tracker.menu()

OUTPUT:

\*\*\*\*\*\*\*\*

Expense Tracker Menu

1. Add Expense

2. View Expenses

3. Analyze Expenses

4. Exit

Enter your choice: 1

Enter amount spent: 600

Enter category (e.g., Food, Transport): food

Enter a brief description: lunch

Expense added successfully!

Expense Tracker Menu

1. Add Expense

2. View Expenses

3. Analyze Expenses

4. Exit

Enter your choice: 2

Recorded Expenses:

1. Date: 2024-11-24, Amount: 600.0, Category: food, Description: lunch

Expense Tracker Menu

1. Add Expense

2. View Expenses

3. Analyze Expenses

4. Exit

Enter your choice: 3

Expense Analysis:

Total Expenses: 600.00

Category-wise Breakdown:

food: 600.00

Expense Tracker Menu

1. Add Expense

2. View Expenses

3. Analyze Expenses

4. Exit

Enter your choice: 4

Thank you for using the Expense Tracker. Goodbye!

=============CODE EXECUTED SUCCESSFULLY============