* Writeup *

Problem statement:

In today's fast-paced world, individuals need to track and manage their expenses effectively. You are tasked with building a personal Expense Tracker that allows to log daily expenses, categorize them, and track spending against a set monthly budget. The tracker should also have the ability to save and load expenses from a file for future reference

Project document को देखने के बाद मेने कुछ steps/ direction बनाई हैं जो इस तरीके से होगी।

Steps to perform:

1) Create an interactive menu with the options for user.
2) Running a Program based on options
3) Create a Program considering each option
   - Create a python Program for add expenses
   - Create a python Program for view expenses
   - Create a python Program for set budget
   - Create a python Program for track budget
   - Create a python Program for save all expenses to a CSV file
   - Create a python Program for load/read expenses to a csv file
4) organize the Program systematically.

Step-1 Create an interactive menu with the option for user.

Add ⑥
Add ④

```
Print('---Menu---')
Print('1. Add Expense')
Print('2. View Expenses')
Print('3. set Budget')
Print('4. Track Budget')
Print('5. Save Expenses')
Print('6. Exit')
Choice = input('choose an option(1-6):')
```

Output

```
--- Menu --
1. Add Expense
2. View Expenses
3. set Budget
4. Track Budget
5. Save Expenses
6. Exit
Choose an option (1-6): 1
```

Step-2 Running ~~a Program~~ this Program based on options

उपरोक्त step मे user जो option चुनता है वही Program run होना चाहिये, जैसेकी user 1. option चुनता है तो Add Expense का Program run होना चाहिये, यानीकी Add Expense का ~~functio~~ function call होना चाहिये।

```
if Choice == '1': add_expense()
elif Choice == '2': view_expenses()
elif Choice == '3': set_budget()
elif Choice == '4': track_budget()
elif Choice == '5': save_expenses()
```

```
elif choice == '6':
    save_expenses()
    print("Exiting the Program.")
else:
    print("invalid option.\n").
```

Add (5)

Output

यदी User 1. option को चुनेगा तो add_expense() program run होगा। 3. option को चुनेगा तो set_budget() program run होगा और User को budget set करनेको कहा जायेगा। option 6 को चुने जानेपर 'Exiting the Program' output आयेगा, यदी User 1 to 6 विकल्पो के अलावा कोइ अन्य input डालने पर ~~invt~~ 'invalid option' बतायेगा।

Step 3 Create a Program considering each option

User के चयनके अनुसार ~~fun~~ function call हो ऐसा Program बनायेंगे। अब हमे इस option को अलगसे Program बनाना है

1) Add an Expenses

```
def add_expense():
    date = input("Enter date (YYYY-MM-DD):")
    category = input("Enter category (e.g. Food, Travel):")
    amount = float(input("Enter amount:"))
    description = input("Enter description:")
    print("Expense added.\n")
add_expense()
```

Add (2)

Output

```
Enter date (YYYY-MM-DD): 2025-09-29
Enter category (eg. food, Travel): travel
Enter amount: 500
Enter description: Junagadh to surat
Expense added.
```

Problem ⇒

~~हमने यहां जो user से in~~

हमने यहां user से जो input लिया है वे हमने ~~save~~ कहीं भी save नहीं किया है मतलब हमने input को कोई ~~variable~~ variable देकर Ream मे save किया नहीं है और आनेवाले Program मे हमे ~~मांगे~~ मांगे हुए input की जरूर पड़ेगी जैसे की view Expenses, set budget आदी मे

variable ये ~~करना~~ ये Problem solve नहीं होगी क्यो की इस Program मे दीया गया variable local होगा हमे Global चाहीये इसलिये हमे ~~~~ variable को पुरे ~~Program~~ Program मे जरूर रखना होगा जिससे ~~सब~~ Program इस variable मे रखी गई ~~inf~~ information को access कर सके.

① expenses=[] अभी हम list data type का use करेंगे ~~क्योकी~~ और कौंस को खाली छोड़ेंगे क्योकी हम append का use करके इसमे user input ~~डाल~~ डालेंगे हमे यहा list data type का uses इसलिये कीया क्योकी इसमे हम Data को बदल सकते है और डुप्लीकेट value Add कर सकते है ये ~~दो~~ (एक साथ) सुविधा हमे नाही tuple मे मिलती है और ~~ना~~ ना तो set मे ~~मिलती~~ है

~~इस~~ उपरोक्त code मे हमे नीचे दी गई line को Add करना पड़ेगा

```
② expenses.append({"date": date, "category": category,
                  "amount": amount, "discription": description})
```

यहीये हमने expenses list मे data को dictionary type मे save कीया है क्योकी आनेवाले Program मे ये data को fetch करना होगा तब हमे Indexing करना आसान होगा

2) View Expenses

```
def View_expenses():
    if not expenses:
        Print('No expenses Recorded.\n')
        return
```

Out Put

User अगर option 1 को छोड़कर 2 direct 2 को चुनता है तब और expenses=[] variable में कोई data नहीं है तो 'No expenses recorded Print होगा और ~~Program return~~ expenses से Program return होगा।

~~अगर ऐसा~~ अगर data है तब

```
print ('your Expenses:')
for exp in expenses:
    if all (k in exp for k in ('date','category','amount',
                               'description')):
        print (f'{exp['date']} | {exp['category']} | ₹{exp
        ['amount']} | {exp['description']}')
print()
```

Out Put

```
Your Expenses
2025-04-24 | travel | ₹500 | Junagadh to Surat
```

हमने यहा पे all() function का use किया है all function में दीये गये Parameter/~~value~~ key expenses में ~~देखेंगे~~ दीयेगये Data के साथ match नहीं होता है तो false होगा यदी match हुआ तब Print में दीयेगये formate के अनुसार expenses में ~~रखा~~ हुआ Data present होगा।

3) set budget

```
def set_budget():                                    Add ——— (3)
    monthly_budget = float(input('Set your monthly
                                 budget: ₹'))
    print('Budget set.\n')
```

~~Out Put~~ Out Put

```
Set your monthly budget: ₹30,000
Budget set.
```

4) Track budget

```
def track_budget():
    total_expense = sum(exp['amount'] for exp in
                        expenses)
    Print(f'Total expenses so far: ₹{total_expense}')
    if total_expense > monthly_budget:
        print('Warning: you have exceeded
              your budget!')
    else:
        print(f'Remaining budget: ₹{monthly_budget -
                                    total_expense}')
    print()
```

Problem ⇒

उपरोक्त code को Run करने से दिक्कत आयेगी क्योंकी monthly_budget variable local variable है। जो हमने पीछले Program मे use किया था monthly_budget को global बनाने केलये हमे पीछले वाले Program मे

③ global monthly_budget line add करनी होगी

Output

```
Total expenses so far: ₹500
Remaining budget: ₹2500
```

5) Save Expenses

```
import csv
def save_expenses():
    with open('expenses.csv', 'w', newline='', encoding='utf8')
        as f:
        writer = csv.DictWriter(f, fieldnames=['date', 'category',
                 'amount', 'description'])
        writer.writeheader()
```

for CSV file

Column केलये

```
for exp in expenses:
    writer.writerow(exp)
print('Expenses saved to csv.\n')
```
row के लिये

Out Put

Expenses saved to csv.

5) Load Expenses():

```
try:
    with open('expenses.csv', 'r', encoding='utf-8') as f:
        reader = csv.DictReader(f)
        for row in reader:
            expenses.append({
                'date': row['date'],
                'category': row['category'],
                'amount': float(row['amount']),
                'discription': row['discription']
            })
    print('Previous expenses loaded.\n')
except FileNotFoundError:
    print('No previous expense data found.\n')
```

Step-4 organize the Program systematically

जबतक user Program से exit होना ना चाहे तबतक Menu show होते रहना चाहीये इसलीये हमे while loop का उपयोग करना होगा

④ While True:

loop से exit होने के लिये

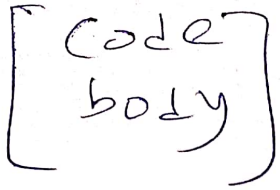
⑤ break

Program आगे बढ़े उससे पहले अगर expenses.csv file में data है तो उसे expenses=[] खाली चल में load करना होगा

(6) load_expenses() function को call करना होगा।

सब फ़ोड को का सही क्रमांक

```
import csv
expenses=[]
monthly_budget=0
def add_expense():
    [ code
      body ]
def view_expenses():
    [ code
      body ]
def set_budget():
    [ code
      body ]
def track_budget():
    [ code
      body ]
def save_expenses():
    [ code
      body ]
def load_expenses():
    [ code
      body ]
def menu():
    load_expenses()
    while True:
```




```
if __name__ == '__main__':
    menu()
```