In [61]:

```python
# import datetime for validation
from datetime import datetime

# import csv
import csv

expenses = [] # Create a list to store expenses
monthly_budget = 0.0 # Monthly budget input
filename = "expenses.csv" # CSV file for saving and loading
```

In [62]:

```python
def add_expense():
    print("=== Add New Expense ===")

    while True:
        date_input = input("Enter the date (YYYY-MM-DD): ").strip()
        try:
            datetime.strptime(date_input, "%Y-%m-%d")
            date = date_input
            break
        except ValueError:
            print("Invalid date format! Please enter in YYYY-MM-DD format")

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

    try:
        amount = float(input("Enter the amount spent: ").strip())
    except ValueError:
        print("Invalid amount! Please enter a number.")
        return

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

    # Create a dictionary with the entered information
    expense = {
        "date": date,
        "category": category,
        "amount": amount,
        "description": description
    }

    expenses.append(expense)
    print("Expense added Successfully!")
```

In [67]:

```python
add_expense()
```

```
=== Add New Expense ===
Expense added Successfully!
```

In [68]:

```python
print("\n Current Expenses:");
for each_expense in expenses:
    print(each_expense)
```

```
 Current Expenses:
{'date': '2025-06-23', 'category': 'Food', 'amount': 304.5, 'description': 'Coffee'}
{'date': '2024-03-05', 'category': 'Travel', 'amount': 1032.0, 'description': 'Chenn
ai'}
```

In [69]:

```python
def view_expenses():
    print("\nView Expense List")

    if not expenses:
        print("No Expenses to display");
        return

    for index, expense in enumerate(expenses, start=1):

        if "date" not in expense or not expense["date"]:
            print(f" Expense #{index} is missing date. Skipping...")
            continue
        if "category" not in expense or not expense["category"]:
            print(f" Expense #{index} is missing category. Skipping...")
            continue

        if "amount" not in expense or not expense["amount"]:
            print(f" Expense #{index} is missing amount. Skipping...")
            continue

        if "description" not in expense or not expense["description"]:
            print(f" Expense #{index} is missing description. Skipping...")
            continue

        print(f"\nExpense #{index}")
        print(f"Date       : {expense['date']}")
        print(f"Category   : {expense['category']}")
        print(f"Amount     : ₹{expense['amount']:.2f}")
        print(f"Description: {expense['description']}")
```

In [70]:

```python
view_expenses()
```

```
View Expense List

Expense #1
Date       : 2025-06-23
Category   : Food
Amount     : ₹304.50
Description: Coffee

Expense #2
Date       : 2024-03-05
Category   : Travel
Amount     : ₹1032.00
Description: Chennai
```

In [76]:

```python
def set_budget():
    global monthly_budget
    print("Set Monthly Budget")
    try:
        monthly_budget = float(input("Enter your total monthly budget:").strip())
        print(f"Monthly budget set to {monthly_budget:.2f}")
    except ValueError:
        print("Invalid input. Please enter a number")

def calculate_total_expenses():
    total = 0.0
    for each_expense in expenses:
        if "amount" in each_expense:
            total += float(each_expense["amount"])
    return total

def track_budget():
    print("Budget Status")
    if monthly_budget == 0:
        print("Set your monthly budget")
        return

    total_spent = calculate_total_expenses()
    print(f" Total expenses so far: ₹{total_spent:.2f}")

    remaining = monthly_budget - total_spent
    if remaining < 0:
        print(f"Exceeded the budget by ₹{abs(remaining):.2f}!")
    else:
        print(f"You have ₹{remaining:.2f} left for the month")
```

In [84]:

```
def save_expenses_to_file():
    try:
        with open(filename, "w", newline='', encoding="utf-8") as file:
            writer = csv.DictWriter(file, fieldnames=["date", "category", "amount",
            writer.writeheader()
            writer.writerows(expenses)
            print(f"Expenses saved to {filename}")
    except Exception as e:
        print(f"Error saving file: {e}")

def load_expenses_from_file():
    try:
        with open(filename, "r", newline='', encoding="utf-8") as file:
            reader = csv.DictReader(file)
            for row in reader:
                try:
                    row["amount"] = float(row["amount"])
                    expenses.append(row)
                except ValueError:
                    continue  # skip if amount is invalid
        print(f"Loaded expenses from {filename}")
    except FileNotFoundError:
        print("No saved CSV file found")
    except Exception as e:
        print(f"Error loading file: {e}")
```

In [ ]:

```
def show_menu():
    while True:
        print("\n=== Personal Expense Tracker Menu ===")
        print("1. Add Expense")
        print("2. View Expenses")
        print("3. Set & Track Budget")
        print("4. Save Expenses to File")
        print("5. Exit")

        choice = input("Choose an option (1-5): ").strip()

        if choice == "1":
            add_expense()
        elif choice == "2":
            view_expenses()
        elif choice == "3":
            set_budget()
            track_budget()
        elif choice == "4":
            save_expenses_to_file()
        elif choice == "5":
            save_expenses_to_file()
            print("Exiting")
            break
        else:
            print("Invalid choice. Please enter a number between 1 and 5.")


# -------------------------
# Start the Project
# -------------------------
load_expenses_from_file()
show_menu()
```

```
No saved CSV file found

=== Personal Expense Tracker Menu ===
1. Add Expense
2. View Expenses
3. Set & Track Budget
4. Save Expenses to File
5. Exit
```

```
View Expense List

Expense #1
Date       : 2025-06-23
Category   : Food
Amount     : ₹304.50
Description: Coffee

Expense #2
Date       : 2024-03-05
Category   : Travel
Amount     : ₹1032.00
Description: Chennai

Expense #3
Date       : 2025-06-23
Category   : Food
Amount     : ₹300.00
Description: Coffee

Expense #4
Date       : 2025-06-25
Category   : Travel
Amount     : ₹799.00
Description: Chennai

=== Personal Expense Tracker Menu ===
1. Add Expense
2. View Expenses
3. Set & Track Budget
4. Save Expenses to File
5. Exit
Set Monthly Budget
Monthly budget set to 6000.00
Budget Status
 Total expenses so far: ₹2435.50
You have ₹3564.50 left for the month

=== Personal Expense Tracker Menu ===
1. Add Expense
2. View Expenses
3. Set & Track Budget
4. Save Expenses to File
5. Exit
Expenses saved to expenses.csv

=== Personal Expense Tracker Menu ===
1. Add Expense
2. View Expenses
3. Set & Track Budget
4. Save Expenses to File
5. Exit
```

In [ ]: