In [50]:

```
# Import necessary modules
from datetime import datetime  # For validating date input
import csv  # For reading and writing CSV files

# Initialize expense tracking variables
expense_records = []          # Stores all logged expense items
monthly_budget = 0.0          # Monthly budget input
expense_data_file = "expenses_data.csv"  # CSV file used to persist expense data
```

In [51]:

```
def add_expense():
    print("=== Add New Expenses ===")

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

    expense_category = input("Enter the expense category (e.g. Travel, Accomodation

    # Keep asking for amount until a valid float is entered
    while True:
        try:
            expense_amount = float(input("Enter the expense amount: ").strip())
            break
        except ValueError:
            print("Invalid amount! Please enter a number.")

    expense_description = input("Enter a brief description of the expense: ").strip

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

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

In [52]:

```
def view_expenses():
    print("\n=== View Expense List ===")

    if not expense_records:
        print("No expenses to display.")
        return

    for index, expense in enumerate(expense_records, 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 [53]:

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

def calculate_total_expenses():
    total = 0.0
    for expense_item in expense_records:
        if "amount" in expense_item and expense_item["amount"]:
            total += float(expense_item["amount"])
    return total

def track_budget():
    print("=== Budget Tracker ===")
    if monthly_budget == 0:
        print("Please set your monthly budget first.")
        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"⚠ Budget exceeded by ₹{abs(remaining):.2f}!")
    else:
        print(f"✅ You have ₹{remaining:.2f} left for the month.")
```

In [54]:

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

def load_expenses_from_file():
    try:
        with open(expense_data_file, "r", newline='', encoding="utf-8") as file:
            reader = csv.DictReader(file)
            for row in reader:
                try:
                    row["amount"] = float(row["amount"])
                    expense_records.append(row)
                except ValueError:
                    continue  # skip if amount is invalid
        print(f"Loaded expenses from {expense_data_file}")
    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
=== Add New Expenses ===
Expense added Successfully!

=== 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 ₹10000.00
=== Budget Tracker ===
Total expenses so far: ₹1000.00
✅ You have ₹9000.00 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
=== Add New Expenses ===
Expense added Successfully!

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

=== 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        : 2025-01-01
Category    : Travel
Amount      : ₹1000.00
Description : Bike ride

Expense #2
Date        : 2025-02-10
Category    : Food
Amount      : ₹1200.00
Description : Dinning

Expense #3
Date        : 2025-03-09
Category    : Accomodation
Amount      : ₹2000.00
Description : Hotel stays

=== 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_data.csv

=== 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 ₹10000.00
=== Budget Tracker ===
Total expenses so far: ₹4200.00
✅ You have ₹5800.00 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_data.csv

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

In [ ]: