```
import datetime

def get_expense_details():

  category = input("Enter expense category: ")
  amount = float(input("Enter expense amount: "))
  description = input("Enter expense description: ")

  return {
      "category": category,
      "amount": amount,
      "description": description
  }

expenses = []

num_expenses = int(input("Enter the number of expenses to add: "))
for _ in range(num_expenses):
  expense_details = get_expense_details()
  expense_details['date'] = datetime.date.today().isoformat()
  expenses.append(expense_details)


print(expenses)
```

```
Enter the number of expenses to add: 4
Enter expense category: Rent
Enter expense amount: 27000
Enter expense description: Monthly rent
Enter expense category: Food
Enter expense amount: 15000
Enter expense description: Total food Budget
Enter expense category: Emis and Bills
Enter expense amount: 28000
Enter expense description: Electricity bill and car emi
Enter expense category: Misc
Enter expense amount: 10000
Enter expense description: Veges,dairy,etc
[{'category': 'Rent', 'amount': 27000.0, 'description': 'Monthly rent', 'date': '2024-11-14'}, {'category': 'Food', 'amount': 15000
```

```
def display_expenses(expenses):

  for expense in expenses:
    print(expense)

display_expenses(expenses)
```

```
{'date': '2024-11-14', 'category': 'Rent', 'amount': 27000.0, 'description': 'Monthly rent'}
{'date': '2024-11-14', 'category': 'Food', 'amount': 15000.0, 'description': 'Total food Budget'}
{'date': '2024-11-14', 'category': 'Emis and Bills', 'amount': 28000.0, 'description': 'Electricity bill and car emi'}
{'date': '2024-11-14', 'category': 'Misc', 'amount': 10000.0, 'description': 'Veges,dairy,etc'}
```

```
def get_monthly_budget():

  while True:
    try:
      budget = float(input("Enter your monthly budget: "))
      if budget > 0:
        return budget
      else:
        print("Invalid input. Budget must be a positive number.")
    except ValueError:
      print("Invalid input. Please enter a number.")

# Example usage:
monthly_budget = get_monthly_budget()
print(f"Your monthly budget is: {monthly_budget}")
```

```
Enter your monthly budget: 90000
Your monthly budget is: 90000.0
```

```
def calculate_expenses_and_balance(expenses, budget):
  total_expenses = sum(expense['amount'] for expense in expenses)

  if total_expenses <= budget:
    remaining_balance = budget - total_expenses
    print(f"Total budget: {monthly_budget}")
    print(f"Total expenses: {total_expenses}")
```

```
    print(f"Remaining balance: {remaining_balance}")
  else:
    print(f"Total expenses: {total_expenses}")
    print("You have exceeded your budget.")


calculate_expenses_and_balance(expenses, monthly_budget)
```

```
Total budget: 90000.0
Total expenses: 80000.0
Remaining balance: 10000.0
```

```
import csv

def save_expenses_to_csv(expenses, filename="expenses.csv"):
  with open(filename, 'w', newline='') as csvfile:
    fieldnames = ['date', 'category', 'amount', 'description']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

    writer.writeheader()
    writer.writerows(expenses)

  print(f"Expenses saved to {filename}")


save_expenses_to_csv(expenses)
save_expenses_to_csv(expenses, "my_expenses.csv")
```

```
Expenses saved to expenses.csv
Expenses saved to my_expenses.csv
```

```
def load_expenses_from_csv(filename="expenses.csv"):

  expenses = []
  with open(filename, 'r', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
      # Convert amount to float
      row['amount'] = float(row['amount'])
      expenses.append(row)
  return expenses
expenses = load_expenses_from_csv()
expenses
```

```
[{'date': '2024-11-14',
  'category': 'Rent',
  'amount': 27000.0,
  'description': 'Monthly rent'},
 {'date': '2024-11-14',
  'category': 'Food',
  'amount': 15000.0,
  'description': 'Total food Budget'},
 {'date': '2024-11-14',
  'category': 'Emis and Bills',
  'amount': 28000.0,
  'description': 'Electricity bill and car emi'},
 {'date': '2024-11-14',
  'category': 'Misc',
  'amount': 10000.0,
  'description': 'Veges,dairy,etc'}]
```

```
def display_menu():
  """Displays a menu of options for the expense tracker."""

  print("\nExpense Tracker Menu:")
  print("1. Add expense")
  print("2. Display expenses")
  print("3. Calculate expenses and balance")
  print("4. Save expenses to CSV")
  print("5. Load expenses from CSV")
  print("6. Exit")

def main():
 expenses = load_expenses_from_csv()
 monthly_budget = 90000.0

 while True:

    display_menu()
    try:
      choice = int(input("Enter your choice (1-6): "))  # integer input
```

```
      if choice == 1:
        expense_details = get_expense_details()
        expense_details['date'] = datetime.date.today().isoformat()
        expenses.append(expense_details)
        print("Expense added successfully!")

      elif choice == 2:
        display_expenses(expenses)
      elif choice == 3:
        calculate_expenses_and_balance(expenses, monthly_budget)
      elif choice == 4:
        save_expenses_to_csv(expenses)
      elif choice == 5:
        expenses = load_expenses_from_csv()
      elif choice == 6:
        break
      else:
        print("Invalid choice. Please enter a number between 1 and 6.")
    except ValueError:
      print("Invalid input. Please enter a number.")

if __name__ == "__main__":
  main()
```

```
Expense Tracker Menu:
1. Add expense
2. Display expenses
3. Calculate expenses and balance
4. Save expenses to CSV
5. Load expenses from CSV
6. Exit
Enter your choice (1-6): 1
Enter expense category: Travel
Enter expense amount: 12000
Enter expense description: Travelling with friends
Expense added successfully!

Expense Tracker Menu:
1. Add expense
2. Display expenses
3. Calculate expenses and balance
4. Save expenses to CSV
5. Load expenses from CSV
6. Exit
Enter your choice (1-6): 2
{'date': '2024-11-14', 'category': 'Rent', 'amount': 27000.0, 'description': 'Monthly rent'}
{'date': '2024-11-14', 'category': 'Food', 'amount': 15000.0, 'description': 'Total food Budget'}
{'date': '2024-11-14', 'category': 'Emis and Bills', 'amount': 28000.0, 'description': 'Electricity bill and car emi'}
{'date': '2024-11-14', 'category': 'Misc', 'amount': 10000.0, 'description': 'Veges,dairy,etc'}
{'category': 'Travel', 'amount': 12000.0, 'description': 'Travelling with friends', 'date': '2024-11-14'}

Expense Tracker Menu:
1. Add expense
2. Display expenses
3. Calculate expenses and balance
4. Save expenses to CSV
5. Load expenses from CSV
6. Exit
Enter your choice (1-6): 3
Total expenses: 92000.0
You have exceeded your budget.

Expense Tracker Menu:
1. Add expense
2. Display expenses
3. Calculate expenses and balance
4. Save expenses to CSV
5. Load expenses from CSV
6. Exit
Enter your choice (1-6): 4
Expenses saved to expenses.csv

Expense Tracker Menu:
1. Add expense
2. Display expenses
3. Calculate expenses and balance
4. Save expenses to CSV
5. Load expenses from CSV
6. Exit
Enter your choice (1-6): 5
```

```
load_expenses_from_csv()
```

```
[{'date': '2024-11-14',
  'category': 'Rent',
  'amount': 27000.0,
  'description': 'Monthly rent'},
 {'date': '2024-11-14',
  'category': 'Food',
  'amount': 15000.0,
  'description': 'Total food Budget'},
 {'date': '2024-11-14',
  'category': 'Emis and Bills',
  'amount': 28000.0,
  'description': 'Electricity bill and car emi'},
 {'date': '2024-11-14',
  'category': 'Misc',
  'amount': 10000.0,
  'description': 'Veges,dairy,etc'},
 {'date': '2024-11-14',
  'category': 'Travel',
  'amount': 12000.0,
  'description': 'Travelling with friends'}]
```