PROJECT-3

K.SOWMYA

CODE:

```
import datetime

class ExpenseTracker:

  def __init__(self):
    self.expenses = {}

  def add_expense(self, category, amount, description):
    date = datetime.date.today()
    if date not in self.expenses:
      self.expenses[date] = []

    self.expenses[date].append({
        'category': category,
        'amount': amount,
        'description': description
    })

  def view_summary(self):
    for date, expenses in self.expenses.items():
      print(f"\nDate: {date}")
      total_expense = 0
      category_expenses = {}

      for expense in expenses:
        total_expense += expense['amount']

        if expense['category'] not in category_expenses:
          category_expenses[expense['category']] = 0

        category_expenses[expense['category']] += expense['amount']
```

```
        print(f"Total Expense: ${total_expense:.2f}")
        print("Category-wise Expenses:")
        for category, amount in category_expenses.items():
            print(f"{category}: ${amount:.2f}")

    def show_menu(self):
        print("\nExpense Tracker Menu:")
        print("1. Add Expense")
        print("2. View Monthly Summary")
        print("3. Exit")

    def run(self):
        while True:
            self.show_menu()
            choice = input("Enter your choice (1/2/3): ")

            if choice == '1':
                category = input("Enter expense category: ")
                amount = float(input("Enter expense amount: "))
                description = input("Enter expense description: ")
                self.add_expense(category, amount, description)
                print("Expense added successfully!")

            elif choice == '2':
                self.view_summary()

            elif choice == '3':
                print("Exiting Expense Tracker. Goodbye!")
                break

            else:
                print("Invalid choice. Please enter 1, 2, or 3.")


if __name__ == "__main__":
    tracker = ExpenseTracker()
```

```
  tracker.run()
'''
We are importing library datetime to get the current date and time.
We are defining a class ExpenseTracker to manage the expenses.
In the __init__ method, we initialize an empty dictionary
self.expenses to store the expenses.
In the add_expense method, we add an expense to the expenses
dictionary.
In the view_summary method, we calculate the total expense for each
date and display it along with the category-wise expenses.
'''
```