Project 3

Expense Tracker

```
import json
from datetime import datetime
import os

# Define the path for storing expense data
DATA_FILE = 'expenses.json'

class ExpensesTracker:
    def _init_(self):
        # Load data from file or initialize a new data structure
        self.expenses = self.load_data()

    def load_data(self):
        """Loads expense data from a JSON file, or initializes an empty list if the file doesn't exist."""
        if os.path.exists(DATA_FILE):
            with open(DATA_FILE, 'r') as file:
                return json.load(file)
        return []

    def save_data(self):
        """Saves expense data to a JSON file."""
        with open(DATA_FILE, 'w') as file:
            json.dump(self.expenses, file, indent=4)

    def add_expense(self, amount, description, category):
        """Adds a new expense to the tracker."""
        expenses = {
```

```
            'date': datetime.now().strftime('%Y-%m-%d'),
            'amount': amount,
            'description': description,
            'category': category
        }
        self.expenses.append(expenses)
        self.save_data()
        print("Expense added successfully!")

    def view_monthly_summary(self, month, year):
        """Displays a summary of expenses for a given month and year."""
        monthly_expenses = [e for e in self.expenses if e['date'].startswith(f'{year}-{month:02d}')]
        total = sum(e['amount'] for e in monthly_expenses)

        print(f"Summary for {year}-{month:02d}:")
        for expense in monthly_expenses:
            print(f"{expenses['date']} - {expenses['category']}: ${expenses['amount']} ({expenses['description']})")
        print(f"Total for {year}-{month:02d}: ${total}")

    def view_category_summary(self, category):
        """Displays a summary of expenses for a given category."""
        category_expenses = [e for e in self.expenses if e['category'].lower() == category.lower()]
        total = sum(e['amount'] for e in category_expenses)

        print(f"Category Summary for '{category}':")
        for expense in category_expenses:
            print(f"{expense['date']} - ${expense['amount']} ({expense['description']})")
        print(f"Total for '{category}': ${total}")
```

```
def run(self):
    """Main method to run the expense tracker user interface."""
    print("Welcome to the Expense Tracker!")
    while True:
        print("\nOptions:")
        print("1. Add Expense")
        print("2. View Monthly Summary")
        print("3. View Category Summary")
        print("4. Exit")

        choice = input("Choose an option (1-4): ")
        try:
            if choice == '1':
                amount = float(input("Enter amount spent: $"))
                description = input("Enter a brief description: ")
                category = input("Enter category (e.g., food, transportation, entertainment): ")
                self.add_expense(amount, description, category)
            elif choice == '2':
                month = int(input("Enter month (1-12): "))
                year = int(input("Enter year (e.g., 2023): "))
                self.view_monthly_summary(month, year)
            elif choice == '3':
                category = input("Enter category to summarize (e.g., food, transportation): ")
                self.view_category_summary(category)
            elif choice == '4':
                print("Exiting Expense Tracker. Goodbye!")
                break
            else:
```

```
                print("Invalid option. Please select a number between 1 and 4.")
            except ValueError:
                print("Invalid input. Please enter numeric values where required.")
            except Exception as e:
                print(f"An error occurred: {e}")

# Entry point to run the program
tracker =ExpensesTracker()
tracker.run()
```

out put:

:\Users\rr175\OneDrive\Desktop\pyscript\pyscript\rak.py'

Welcome to the Expense Tracker!

Options:

1. Add Expense
2. View Monthly Summary
3. View Category Summary
4. Exit

Choose an option (1-4): 1

Enter amount spent: $55000

Enter a brief description: rent payment

Enter category (e.g., food, transportation, entertainment): housing

Options:

1. Add Expense
2. View Monthly Summary
3. View Category Summary
4. Exit

Choose an option (1-4): 2

Enter month (1-12): 5

Enter year (e.g., 2023): 2022

Options:

1. Add Expense

2. View Monthly Summary

3. View Category Summary

4. Exit

Choose an option (1-4): 3

Enter category to summarize (e.g., food, transportation): food

Options:

1. Add Expense

2. View Monthly Summary

3. View Category Summary

4. Exit

Choose an option (1-4): 4

Exiting Expense Tracker. Goodbye!

PS C:\Users\rr175\OneDrive\Desktop\pyscript\pyscript>