Project Objectives:

To accurately outline the scope of work required for a project, it is crucial to first identify its objectives. Pinpointing what the project hopes to accomplish will assist in determining its inclusions and limitations.

User Input and Data Management: Develop a system that allows users to input their daily expenses.

Data Storage: Implement a mechanism to store and manage the entered expense data.

Expense Categories: Categorize expenses into different categories for better organization.

Data Analysis: Provide users with insights into their spending patterns, such as monthly summaries and category-wise expenditure.

User-Friendly Interface: Create a user-friendly interface for a seamless user experience.

Error Handling: Implement error handling to ensure the application can handle unexpected inputs gracefully.

Documentation: Document your code effectively to demonstrate clarity and understanding.

Requirements and Features:

User Input: Allow users to input their daily expenses, including the amount spent and a brief description.

Data Storage: Use appropriate data structures or file handling techniques to store and retrieve expense data.

Expense Categories: Implement the ability for users to categorize their expenses (e.g., food, transportation, entertainment).

Data Analysis: Provide users with the option to view summaries of their monthly expenses and category-wise expenditure.

User Interface: Create a simple and intuitive user interface to interact with the Expense Tracker.

Error Handling: Include error handling mechanisms to address potential issues during user interaction.

Documentation: Provide clear documentation for your code, explaining the logic behind key functions and overall program structure.

Source code:

import json

import os

import datetime

def load\_expense\_data():

if os.path.exists("expense\_data.json"):

with open("expense\_data.json", "r") as file:

return json.load(file)

else:

return {}

def save\_expense\_data(expense\_data):

with open("expense\_data.json", "w") as file:

json.dump(expense\_data, file)

def add\_expense(expense\_data):

date = datetime.datetime.now().strftime("%Y-%m-%d")

amount = float(input("Enter the amount spent: "))

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

category = input("Enter the category of expense: ")

if date not in expense\_data:

expense\_data[date] = []

expense\_data[date].append({"amount": amount, "description": description, "category": category})

print("Expense added successfully!")

def categorize\_expenses(expense\_data):

categories = set()

for date in expense\_data:

for expense in expense\_data[date]:

categories.add(expense["category"])

print("Expense categories:", categories)

def analyze\_expenses(expense\_data):

total\_expense = 0

category\_expenses = {}

for date in expense\_data:

for expense in expense\_data[date]:

total\_expense += expense["amount"]

category = expense["category"]

category\_expenses[category] = category\_expenses.get(category, 0) + expense["amount"]

print("Total expenses:", total\_expense)

print("Category-wise expenses:")

for category, amount in category\_expenses.items():

print(f"{category}: {amount}")

def monthly\_summary(expense\_data):

month = input("Enter the month (MM/YYYY): ")

total\_expense = 0

for date in expense\_data:

if date.startswith(month):

for expense in expense\_data[date]:

total\_expense += expense["amount"]

print(f"Total expenses for {month}: {total\_expense}")

def user\_interface():

expense\_data = load\_expense\_data()

while True:

print("\nExpense Tracker Menu:")

print("1. Add Expense")

print("2. Categorize Expenses")

print("3. Analyze Expenses")

print("4. Monthly Summary")

print("5. Exit")

choice = input("Enter your choice: ")

if choice == "1":

add\_expense(expense\_data)

elif choice == "2":

categorize\_expenses(expense\_data)

elif choice == "3":

analyze\_expenses(expense\_data)

elif choice == "4":

monthly\_summary(expense\_data)

elif choice == "5":

save\_expense\_data(expense\_data)

print("Exiting... Goodbye!")

break

else:

print("Invalid choice. Please try again.")

def main():

user\_interface()

if \_\_name\_\_ == "\_\_main\_\_":

main()