

TERM PROJECT

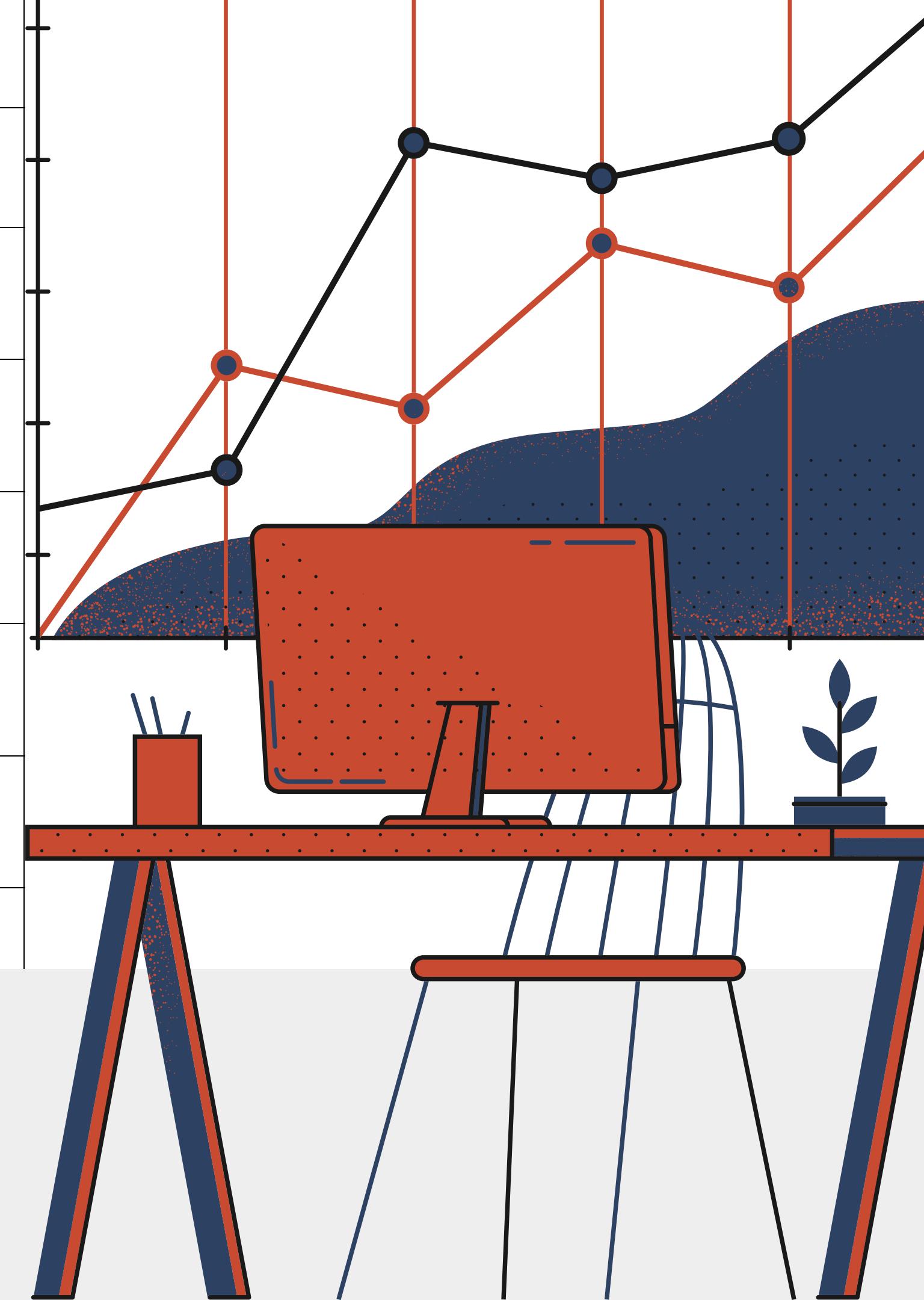
EN051004 Computer Programming
for Automation and Robotics

สมาชิก

643040004-8 นางสาวกัญญาณัฐ ประภาเวสัง

643040022-6 นางสาวปัญญา อุดศักดิ์สอดใส

643040239-1 นางสาวชาคริยา จันทร์คำนิ





README

เกี่ยวกับโปรแกรม

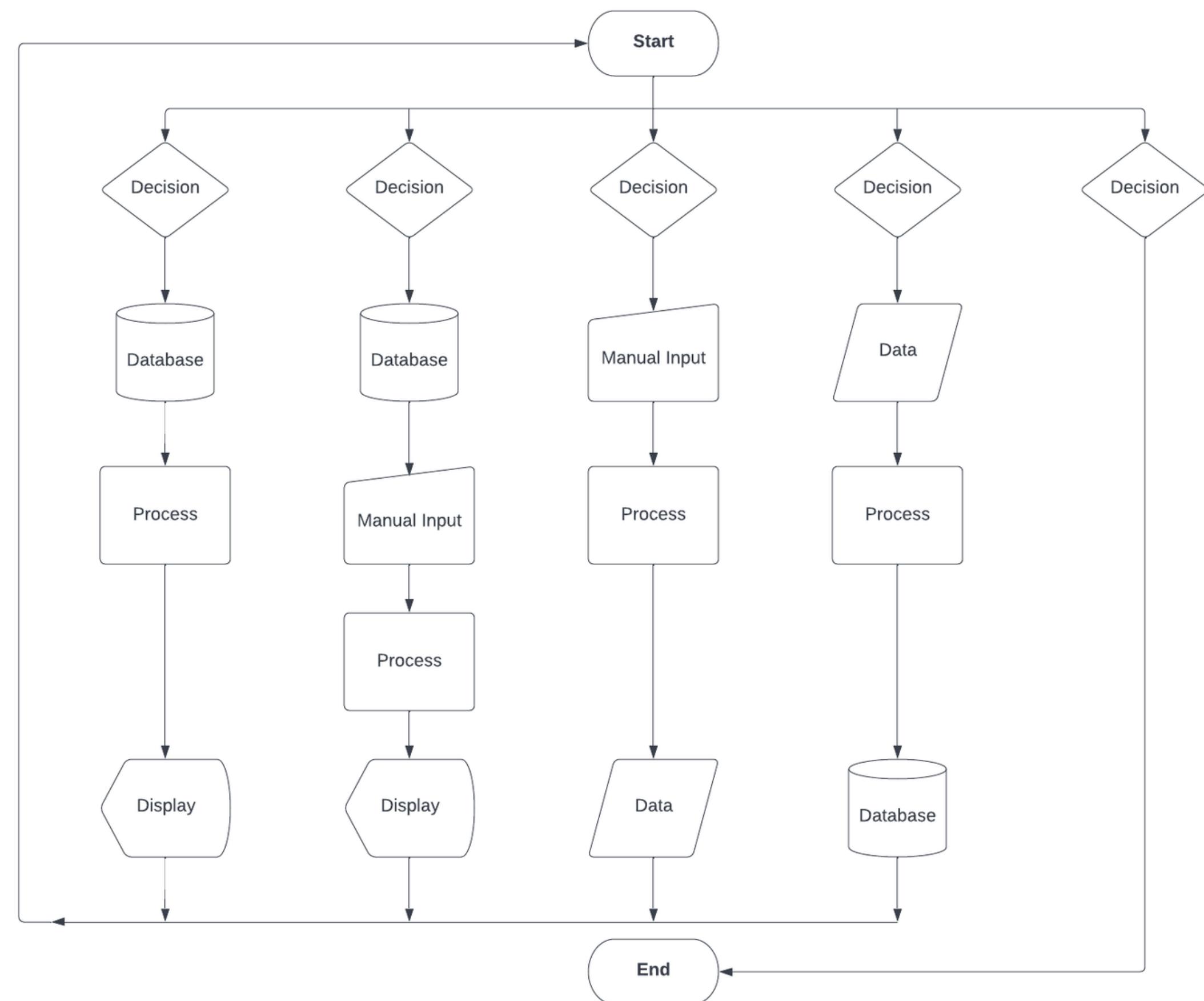
เกี่ยวกับโปรแกรม



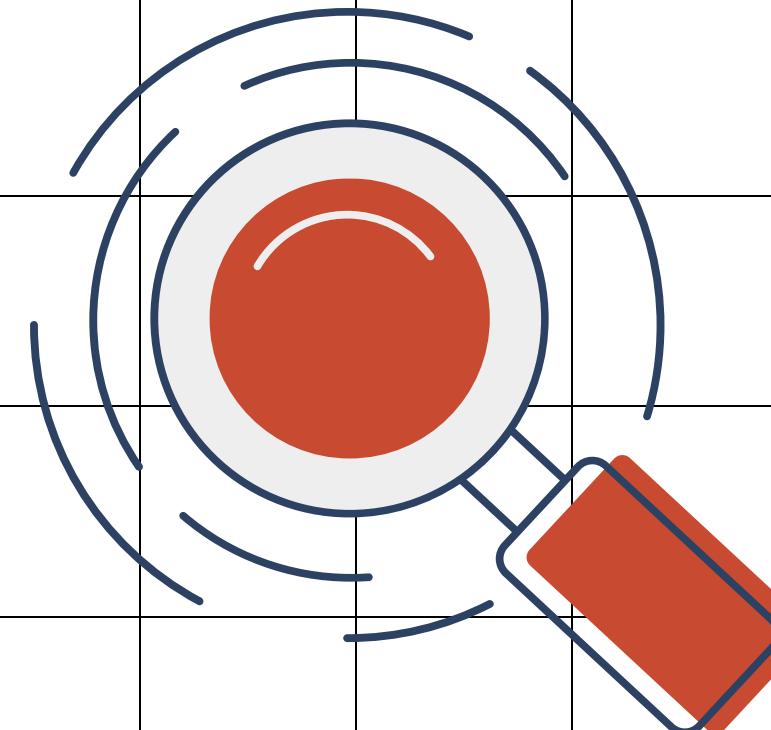
เกี่ยวกับโปรแกรม

- บันทึกรายรับรายจ่ายที่สามารถดูบันทึก ย้อนหลัง หรือดูภาพรวมของการใช้จ่ายได้
- Python based
- มีฟังก์ชันการใช้งานรวมทั้งหมด 4 ฟังก์ชันหลัก

Flowchart



ฟังก์ชันหลัก (Main function)



การบันทึกรายรับรายจ่าย

ในส่วนของบันทึกรายรับรายจ่าย ผู้ใช้งานสามารถระบุ

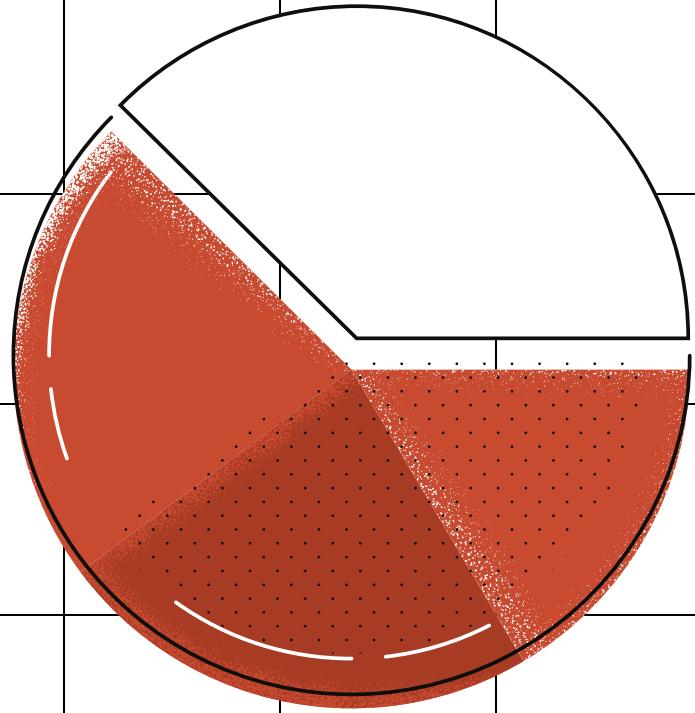
- วันที่ทำธุรกรรม เช่น ปัจจุบัน, ย้อนหลัง
- ประเภทของธุรกรรม เช่น รายรับ, รายจ่าย
- รายละเอียดธุรกรรม เช่น ค่าที่พัก, เงินเดือน
- จำนวนเงิน

ฟังก์ชันหลัก (Main function)

การบันทึก และโหลดข้อมูล

โปรแกรมนี้ผู้ใช้สามารถบันทึกไฟล์ในชื่อที่ต้องการได้ และสามารถอ่านไฟล์ที่เลือกจากที่ได้ก็ได้ในเครื่องไม่จำเป็นต้องอยู่ในโฟลเดอร์เดียวกัน กับโปรแกรม แต่ต้องอยู่ในรูปแบบของไฟล์ .csv

ฟังก์ชันหลัก (Main function)



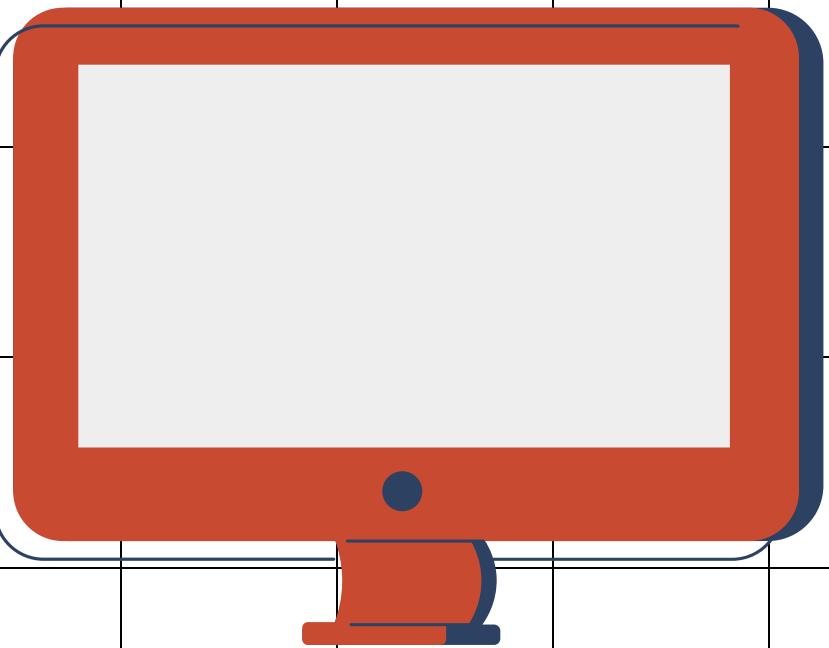
กราฟรายจ่ายในแต่ละเดือน

ในกราฟที่ได้จะมีการระบุ

- ประเภทของสิ่งที่จ่ายไป
- สิ่งที่จ่ายไปนั้นคิดเป็นกี่ % ของรายจ่ายก็คงด

เพื่อกำผู้ใช้งานจะสามารถอ่าน และสังเกตได้ง่ายว่า ใช้จ่ายสิ่งใดมากกว่า หรือน้อยกว่ากัน

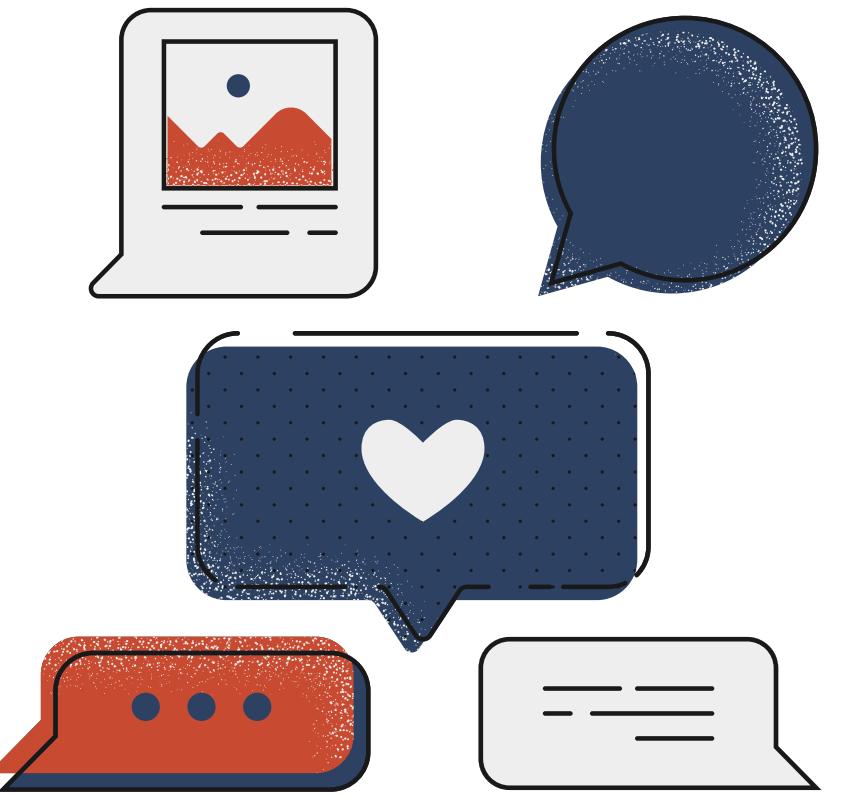
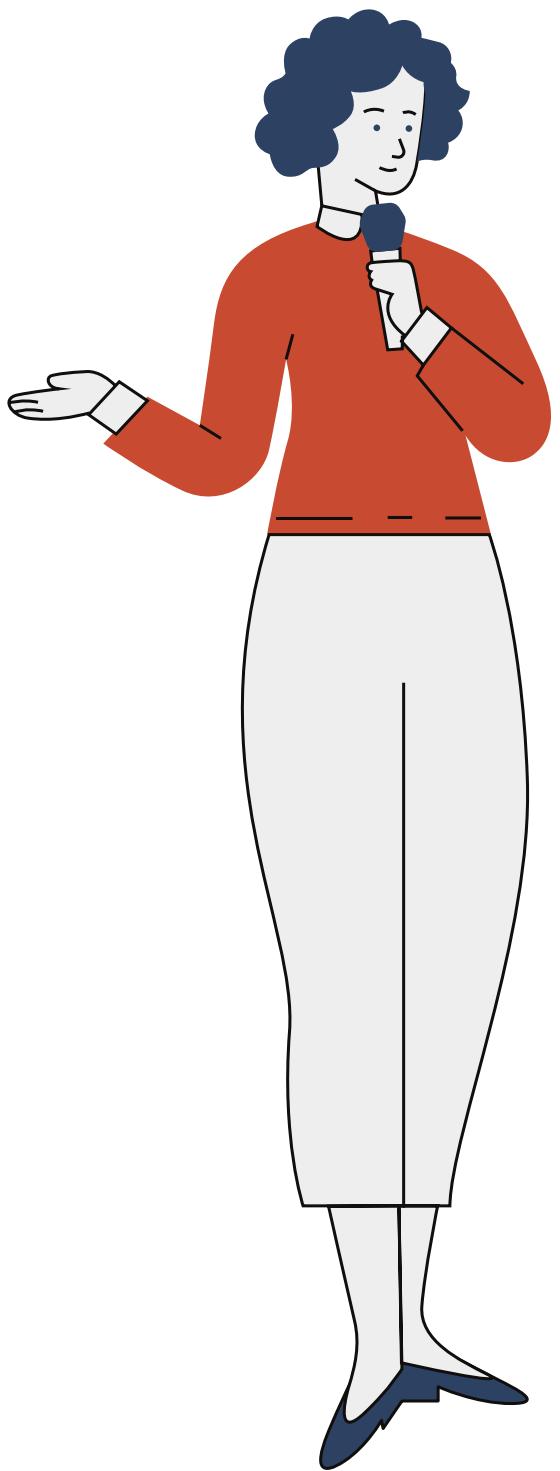
ฟังก์ชันหลัก (Main function)



ประวัติรายรับรายจ่าย
ที่เคยบันทึกไว้ทั้งหมด

ในส่วนของการดูรายการธุรกรรมทั้งหมดที่เคยบันทึกไว้นั้น
โปรแกรมจะอ่านค่าจากไฟล์ .csv และนำมาใช้ในรูปแบบ
dataframe

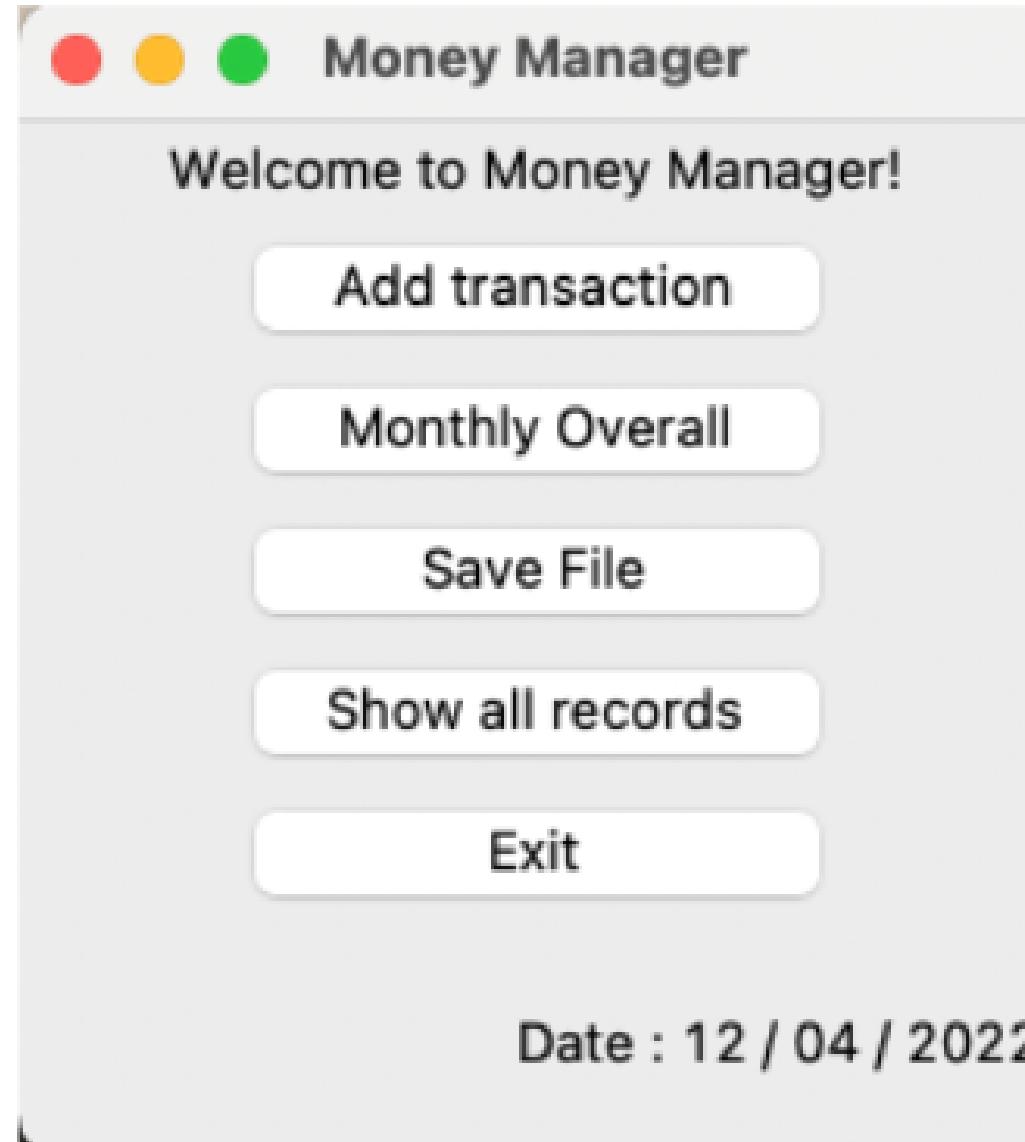
หลังจากนั้นจึงนำมาแสดงผลในรูปแบบตารางบนตัวโปรแกรม



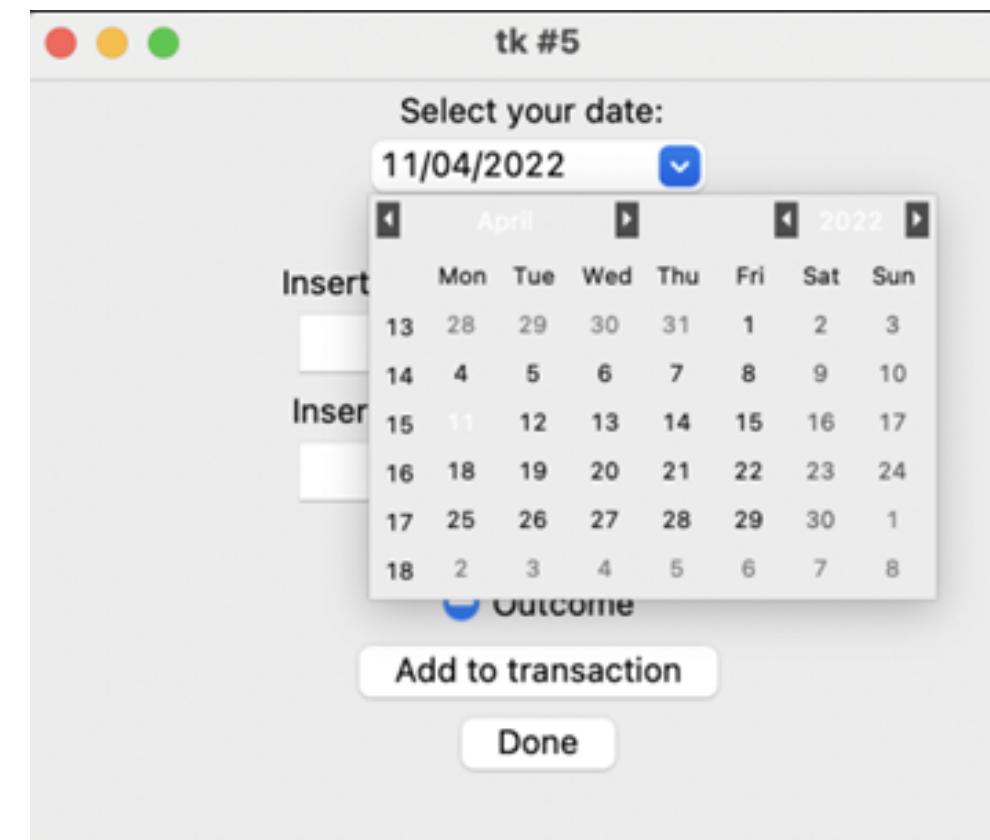
คู่มือการใช้งาน

- การเพิ่มบันทึกรายรับรายจ่าย
- การเข้าดูรายการจ่ายในแต่ละเดือน
- การเข้าดูประวัติรายรับรายจ่ายกั้งหมัด

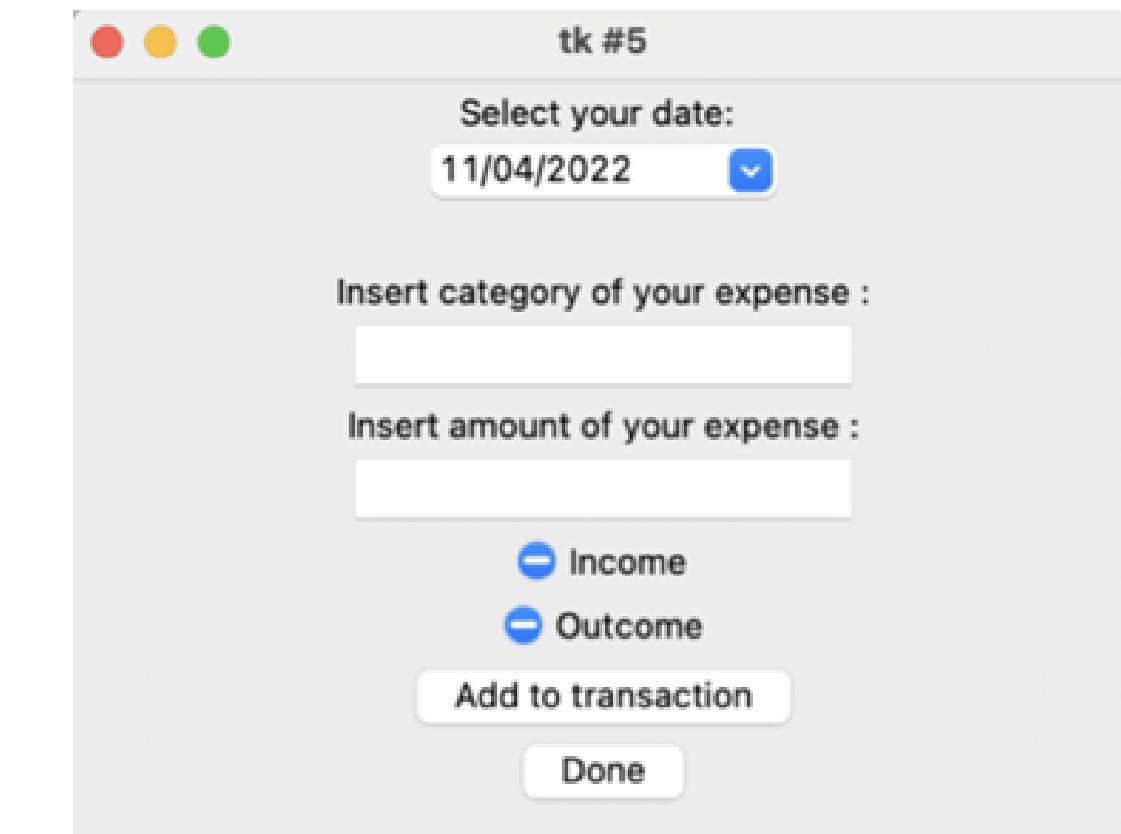
1. การเพิ่มบันทึกรายรับรายจ่าย



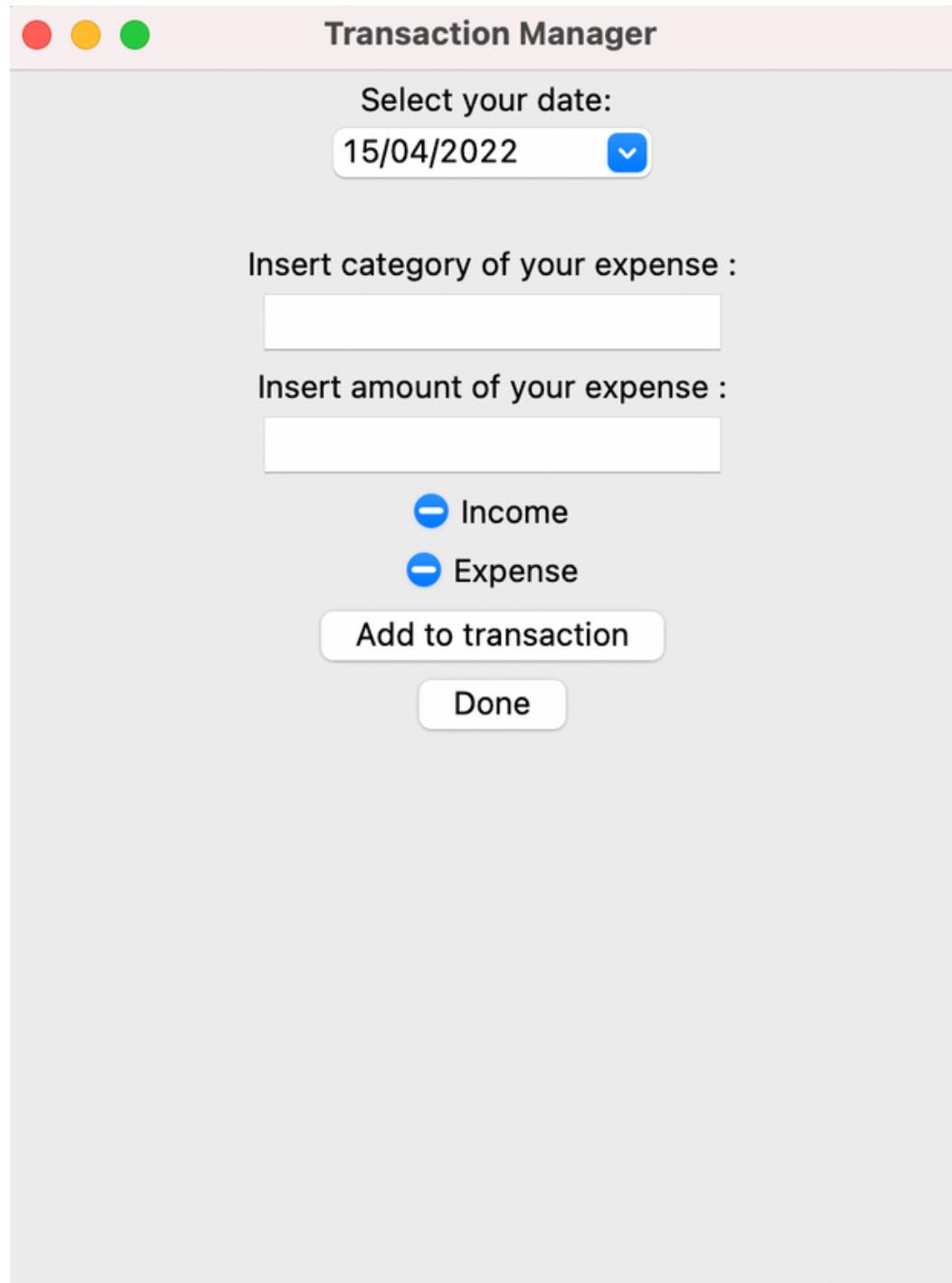
1. เมื่อเปิดหน้าโปรแกรม ขึ้นแรกให้กด Add transaction



2. ให้กรอกข้อมูลวันที่/เดือน/ปี
(ถ้าหากเราไม่ได้กรอกข้อมูล วันที่/เดือน/ปี
ลงไปโปรแกรมจะแสดงวันที่ปัจจุบัน)

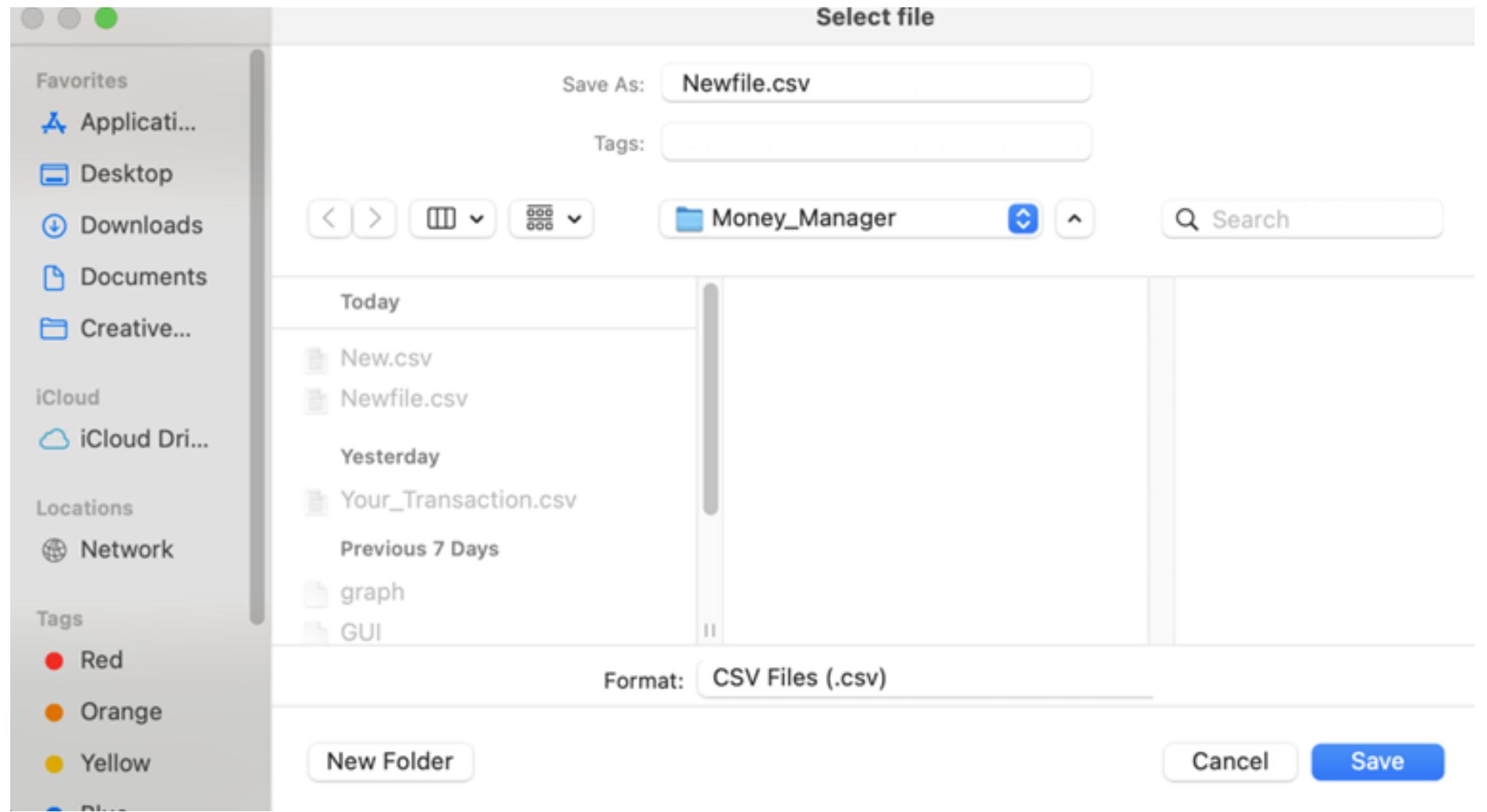


3. กรอกข้อมูลหมวดหมู่ค่าใช้จ่ายของคุณ
และกรอกจำนวนเงินที่คุณใช้จ่าย พร้อมกับ
เลือกประเภทว่าเป็น Income (รายรับ)
หรือ Outcome (รายจ่าย)



**3.1 หากกรอกข้อมูลเสร็จแล้วไม่มีข้อมูลที่ต้องกรอกเพิ่มเติม
ให้กด Add to transaction จากนั้นกด Done
(แต่การกด Done จะไม่มีการบันทึกจนกว่าจะกด Save File)**

**3.2 หากมีข้อมูลที่ต้องการกรอกมากกว่า 1 ครั้ง
ให้กด Add to transaction
แล้วสามารถกรอกข้อมูลใหม่ต่อได้**



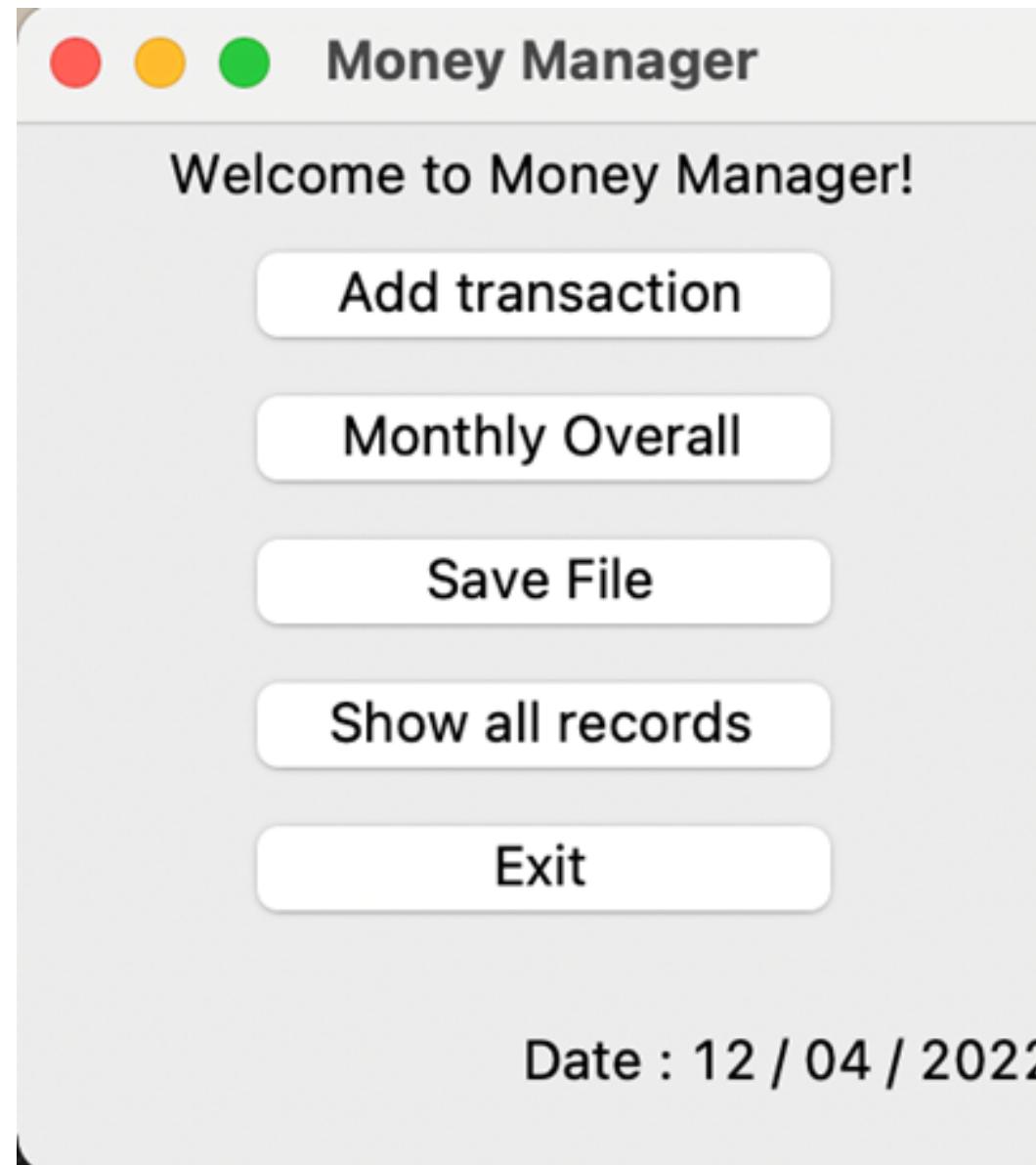
4. หลังจากที่เรากด Done แล้วให้กด Save File

A screenshot of a Tkinter application window titled "tk #4". The window has a light gray header bar with three colored circles (red, yellow, green) on the left. The main area contains a table with a dark gray header row and white body rows. The columns are labeled "Date", "Category", "Expense", and "Type". There are three rows of data:

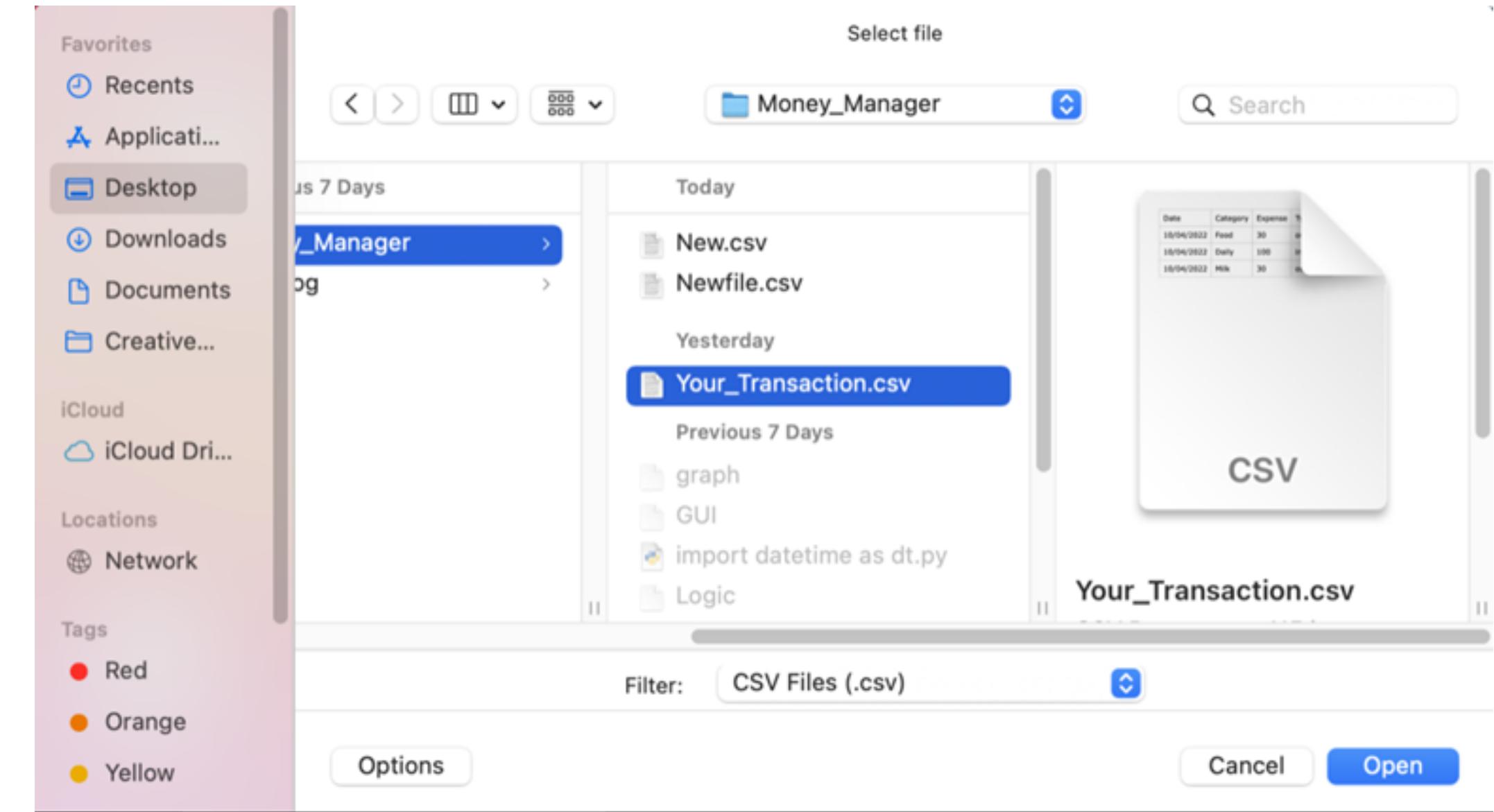
	Date	Category	Expense	Type
1	10/04/2022	Food	30	outcome
2	10/04/2022	Daily	100	income
3	10/04/2022	Milk	30	outcome

5. หลังจากที่เรากด Save จะมีตารางแสดงข้อมูลที่เรากรอก

2. การเข้าดูรายการจ่ายในแต่ละเดือน



1. ให้กด Monthly Overall

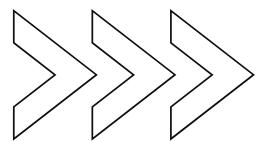


2. เลือกไฟล์ที่เราต้องการ

Overall Spending

Insert month and year in format mm/yyyy

Confirm



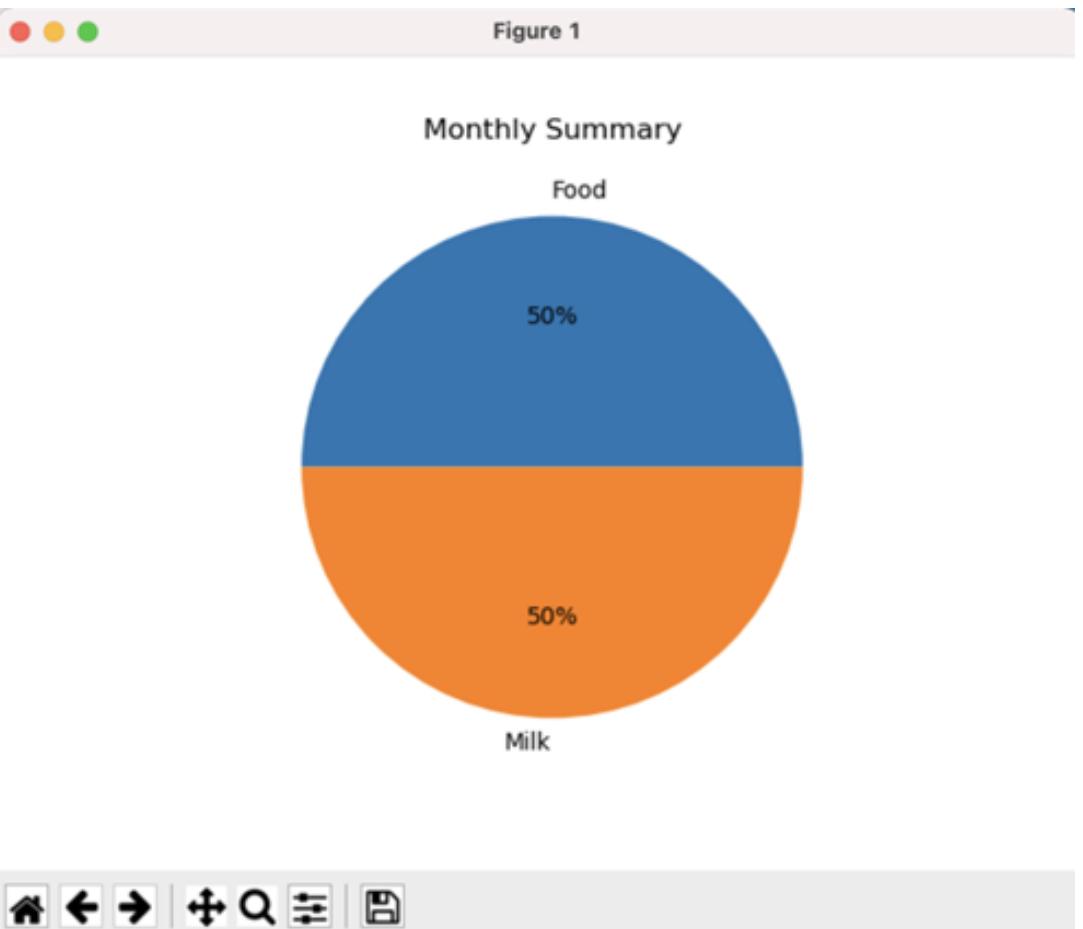
Overall Spending

Insert month and year in format mm/yyyy

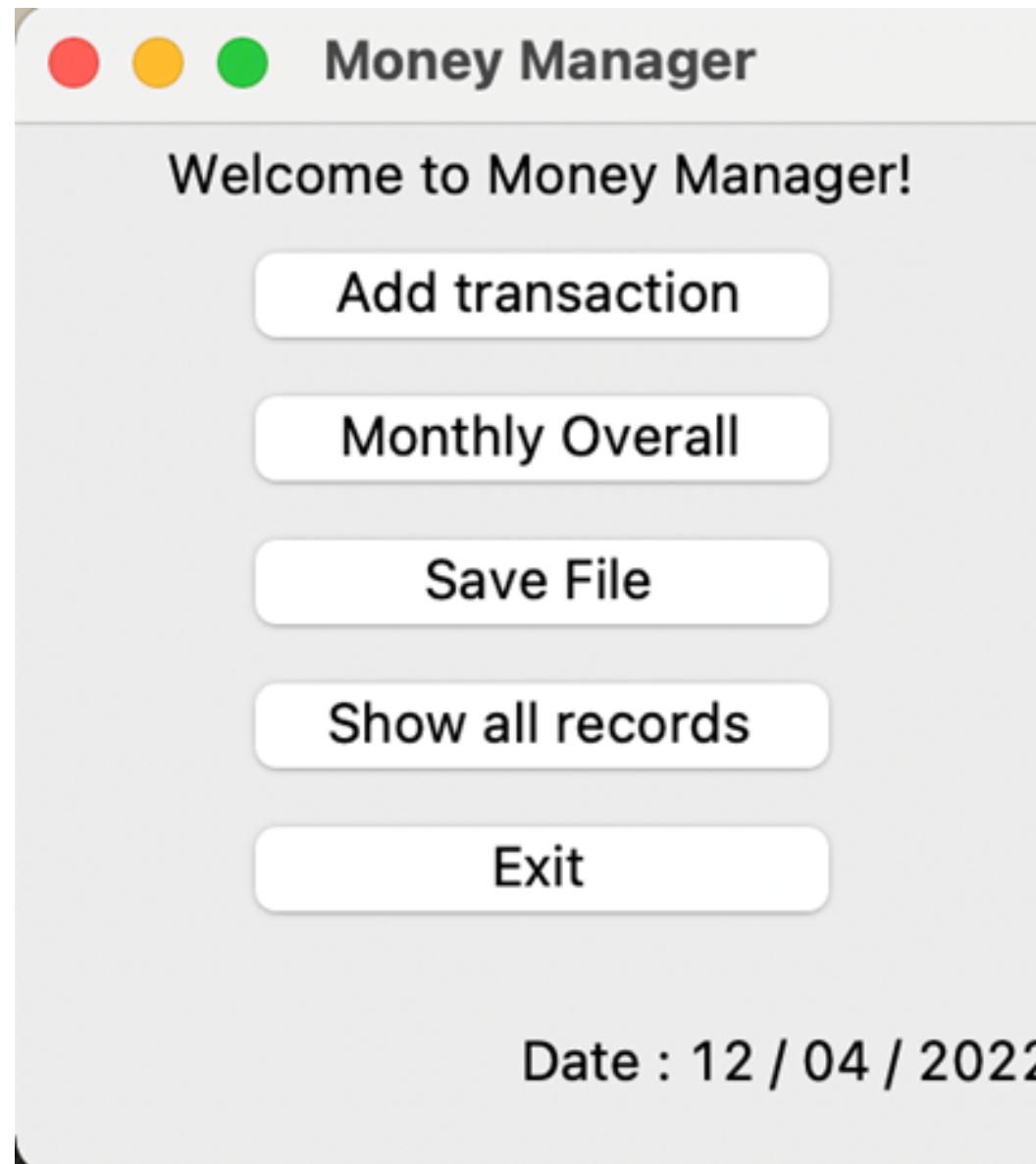
Confirm

3. กรอกเดือน/ปีค.ศ. ที่ต้องการดูกราฟข้อมูล และกด Confirm

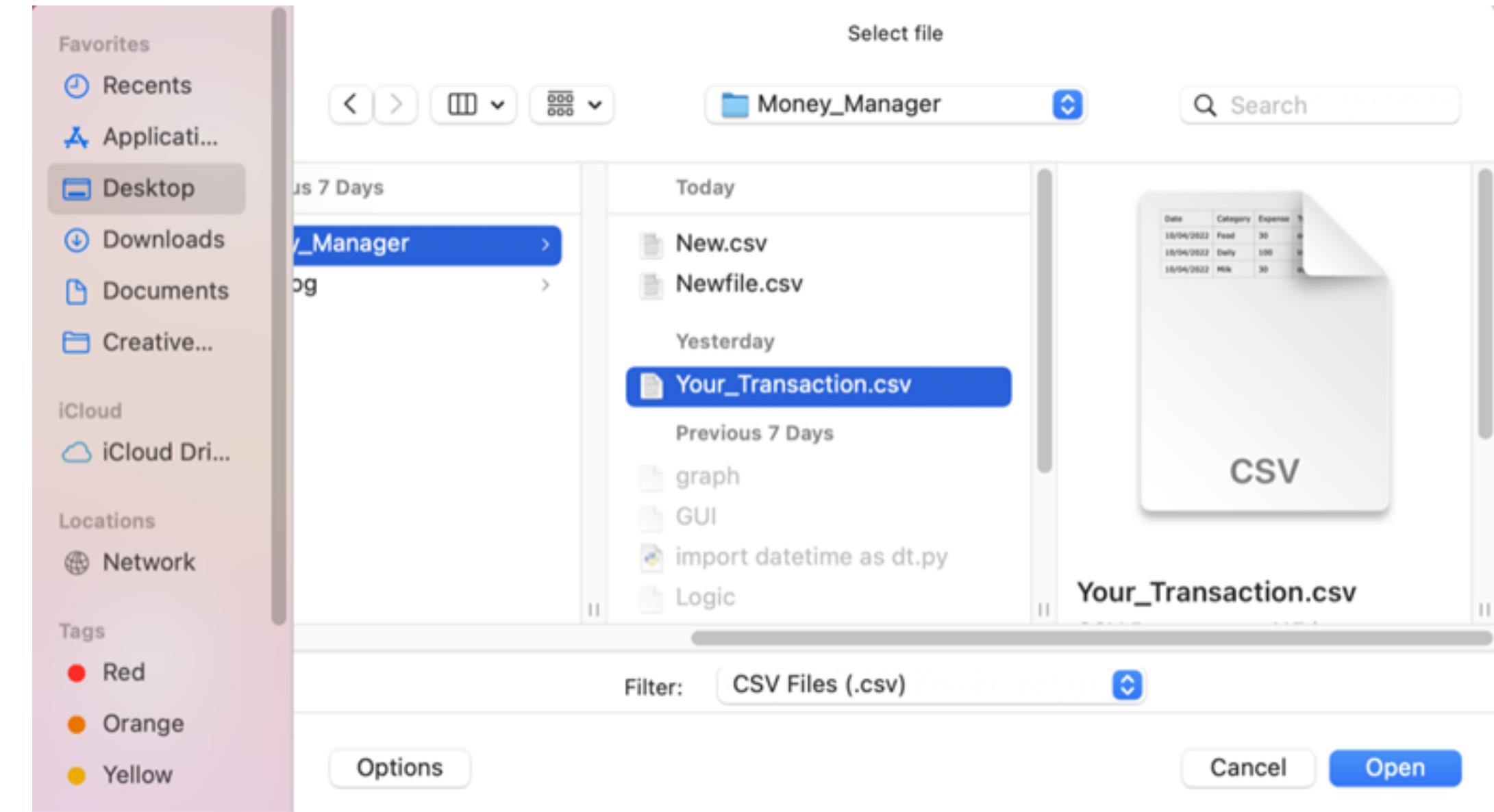
4. กราฟแสดงข้อมูลหมวดหมู่ค่าใช้จ่ายของคุณ
และร้อยละของเงินที่คุณใช้จ่ายไปในเดือนนั้นๆ



3. การเข้าดูประวัติรายรับรายจ่ายทั้งหมด



1. กด Show all records

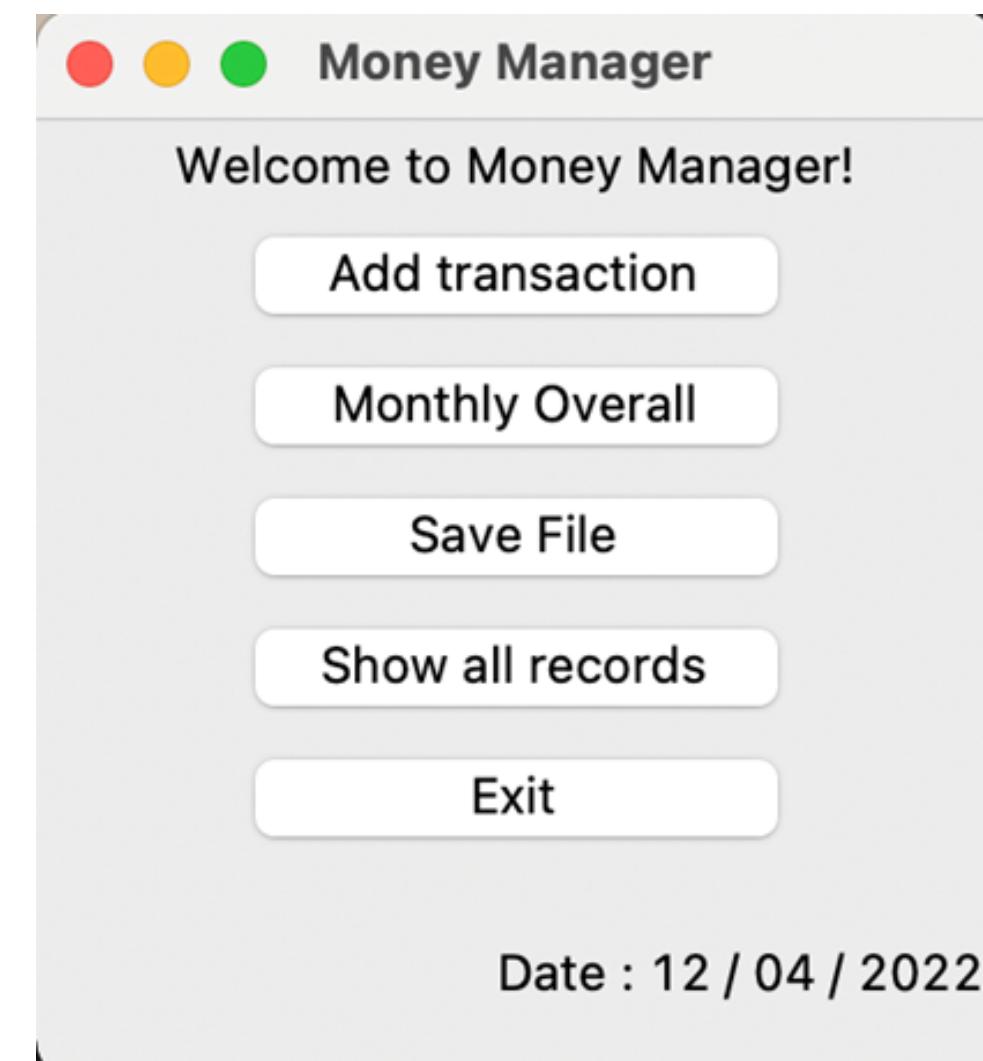


2. เลือกไฟล์ที่เราต้องการ

3. โปรแกรมจะแสดงข้อมูลรายรับรายจ่ายทั้งหมด
ที่ผ่านมาในรูปแบบตาราง

	Date	Category	Expense	Type	
1	10/04/2022	Food	30	outcome	
2	10/04/2022	Daily	100	income	
3	10/04/2022	Milk	30	outcome	

4. หากทำงานเสร็จไม่มีข้อมูลที่ต้องกรอกเพิ่มเติม
หรือแก้ไข กด Exit เพื่้ออกจากหน้าโปรแกรม





Code Preview

Function setup

```
#initial library
from tkinter import *                                     #For Creating GUI
from tkinter import messagebox, filedialog
from tkcalendar import DateEntry
import datetime as dt                                     #For datetime displaying
import pandas as pd                                      #For Creating Dataframe
import pandastable as pt                                 #For Tabel displaying and Graph plotting on GUI
from pandastable import Table
import matplotlib.pyplot as plt
import csv, os                                         #File Writing & Reading, Collect file path

#Main function setup
history = {"Date": [], "Category": [], "Expense": [], "Type": []}      #store data
headers = ["Date", "Category", "Expense", "Type"]                      #Tabel and data header
```

File Writing

```
# name of csv file
save_as = filedialog.asksaveasfilename(title = "Select file", defaultextension=".csv", filetypes = [("CSV Files","*.csv")]
#filename = "Your_Transaction.csv"
if save_as:
    filename = save_as
    with open(filename, "a") as outfile:

        # creating a csv writer object
        writerfile = csv.writer(outfile)
        file_is_empty = os.stat(filename).st_size == 0

        if file_is_empty:
            # writing dictionary keys as headings of csv
            writerfile.writerow(history.keys())

        # writing list of dictionary
        writerfile.writerows(zip(*history.values()))
```

GUI Button setup

```
#Menu Buttons Setup
Label(menu, text="Welcome to Money Manager!").pack()

#Outcome/income button
tran = Button(menu, text = 'Add transaction', bd = '5',width=12,command = transaction)
tran.pack(side = 'top')

#Overall Graph button
graph = Button(menu, text = 'Monthly Overall', bd = '5',width=12,command = graph)
graph.pack(side = 'top')

#Save and Load file button
save = Button(menu, text = 'Save File', bd = '5',width=12,command = save)
save.pack(side = 'top')

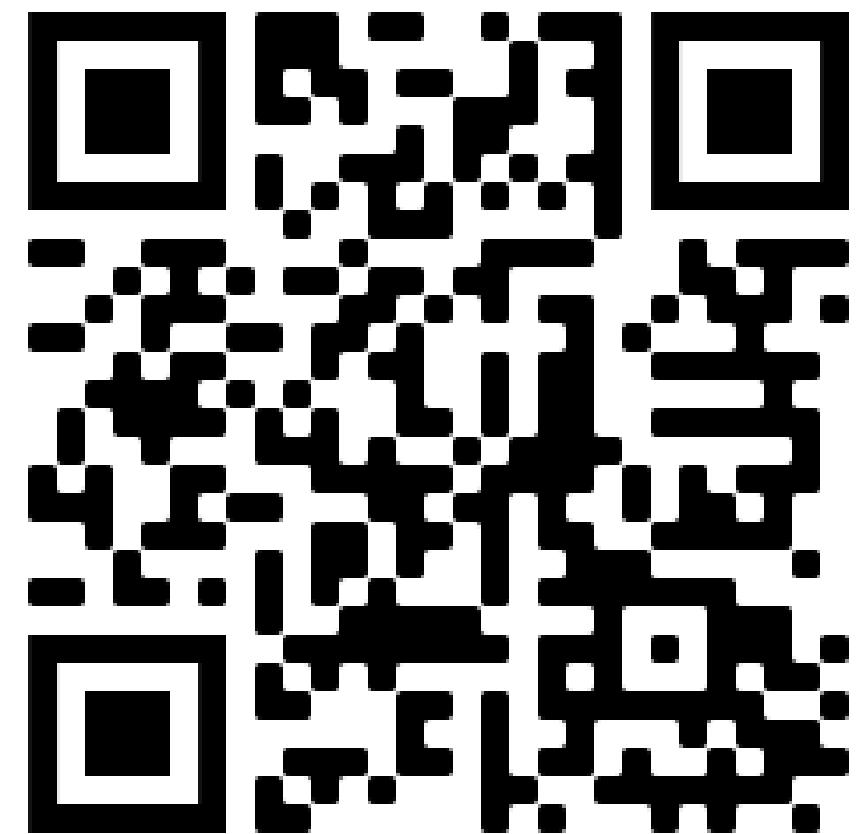
record = Button(menu, text = 'Show all records', bd = '5',width=12,command = all_records)
record.pack(side = 'top')

#Exit button
exit = Button(menu, text = 'Exit', bd = '5',width=12,command = menu.destroy)
exit.pack(side = 'top')
```

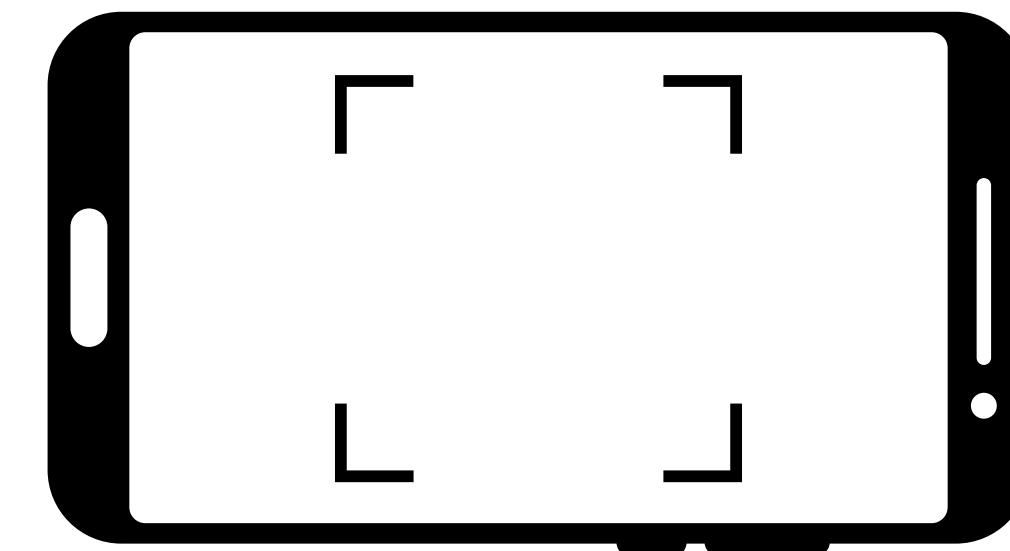
Demo : Money Manager



What's next?



SCAN ME



Our GitHub!