

FINAL PROJECT REPORT

Advanced Computer
Programming

Group 11

- Tsogbat Bat-erdene /113021197/
- Khuslen Gantumur /113021189/

OVERVIEW: LIBRARIES

```
import tkinter as tk
from tkinter import ttk, messagebox, filedialog, simpledialog
from tkcalendar import DateEntry
from database import connect_db, add_transaction, get_all_transactions, get_summary, delete_transaction, \
    get_monthly_summary
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from datetime import datetime
import csv
```

TKINTER

tkinter: Python's built-in standard GUI library used to create desktop applications.

- tk: The base module for widgets like windows, labels, buttons, etc.
- ttk: Themed widgets (modern-looking versions of default Tkinter widgets).
- messagebox: Displays popup dialogs (e.g., alerts, confirmations).
- filedialog: Used for file browsing and saving (e.g., exporting to CSV).
- simpledialog: Prompts the user for input in a dialog box.

Used For: Designing the user interface and handling user interactions.

TKCALENDAR

- DateEntry: A widget that lets users select dates from a calendar.
- Third-party library (needs to be installed via `pip install tkcalendar`).

Used For: Letting users pick the date of a transaction easily.

CUSTOM DATABASE MODULE

This imports functions from the database file database.py.

- These functions handle:
- Connecting to the database.
- Adding, deleting, and reading transaction records.
- Calculating summaries and monthly statistics.

Used For: Separating backend logic and database interaction from the GUI code.

MATPLOTLIB: DATA VISUALIZATION

- `matplotlib.pyplot`: A powerful library to create charts and plots.
- `FigureCanvasTkAgg`: Embeds Matplotlib charts inside Tkinter windows.

Used For: Visualizing expenses via bar charts, pie charts, and other graphs within the app.

DATETIME: DATE AND TIME UTILITIES

- Provides tools for working with dates and times.
- `datetime.now()`, `datetime.strptime()` are commonly used to parse and format dates.

Used For: Converting string inputs into proper date objects and displaying them.

CSV: FILE HANDLING

- Built-in Python library to read and write CSV (Comma Separated Values) files.
- Used to export transactions to a CSV format that users can open in Excel or Google Sheets.

Used For: Export feature – backing up data or sharing reports.

SQLITE3

- Built-in Python library to create and manage a local database.
- Used For: Storing and retrieving all expense records.

PROJECT GOALS

Our project, "Expense Tracker", allows users to input and categorize income and expenses, filter data by month/year/category, and view visual summaries. It offers multiple types of data visualization including pie charts, bar charts, and waterfall plots. It also supports exporting data to CSV .

IMPLEMENTATION

```
1 usage
class ExpenseTracker:
    def __init__(self, root):
        self.root = root
        self.root.title("Expense Tracker")
        self.root.geometry("1200x650")

        self.style = ttk.Style()
        self.categories = ["Food", "Transport", "Bi

        self.login_screen()

1 usage
def login_screen(self):
    password = simpdialog.askstring(title: "Log
```

```
def connect_db():
    conn = sqlite3.connect(DB_NAME)
    c = conn.cursor()
    c.execute("""
        CREATE TABLE IF NOT EXISTS transactions (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            date TEXT NOT NULL,
            category TEXT NOT NULL,
            amount REAL NOT NULL,
            type TEXT NOT NULL
        )
    """)
    conn.commit()
    conn.close()
```

KEY COMPONENTS

Two Main Classes:

- SafeDateEntry: Custom date picker with better error handling
- ExpenseTracker: Main application logic, GUI, and data handling

Database Functions:

Handled separately in database.py (Insert, Read, Update, Delete, Filter, Summarize)

SAFEDATEENTRY CLASS

Purpose:

Custom subclass of DateEntry (from tkcalendar) to prevent focus-related errors

Fields:

- Inherits all fields from DateEntry

Methods:

- `on_focus_out()`: Overrides calendar focus behavior
- Prevents app crashes during rapid tabbing or focus changes

EXPENSETRACKER CLASS

– FIELDS

Main GUI class, connects UI with business logic and database

Main Fields:

- root: Tkinter main window
- tree: Treeview widget (table of transactions)
- categories: Fixed list of transaction categories
- style: Tracks current theme (light/dark)
- Multiple Tkinter widgets (Entry, Combobox, Button, etc.)

EXPENSETRACKER CLASS

– METHODS

UI & Authentication

- `__init__()`: Launches app and login window
- `login_screen()`: Prompts for password using `simpledialog`
- `setup_ui()`: Builds full GUI layout and widgets

Transactions

- `add_transaction()`: Adds validated record to DB
- `refresh_table()`: Reloads data and updates UI
- `remove_transaction()`: Deletes selected item

EXPENSETRACKER CLASS

– METHODS

Data Visualization

- `show_bar_chart()`: Bar chart by category
- `show_progressBars()`: Progress bars for each category
- `show_waterfall_chart()`: Waterfall graph for profit/loss
- `show_pie_chart()`: Pie chart for category distribution
- `show_monthly_chart()`: Monthly income vs. expense

Other Utilities

- `apply_filter()`: Filters transactions by month/year/category
- `export_to_csv()`: Save records to CSV file
- `toggle_theme()`: Switch between light and dark themes
- `on_type_change()`: Auto-selects categories for Income/Expense

DATABASE FUNCTIONS

(FROM DATABASE.PY)

- `connect_db()`: Opens SQLite database, creates table if needed
- `add_transaction(date, category, amount, type)`: Inserts a row
- `get_all_transactions(...)`: Retrieves all/filtered rows
- `get_summary()`: Returns income and expense totals
- `delete_transaction(id)`: Deletes record by ID
- `get_monthly_summary()`: Summarizes monthly data for charts

RESULT

Expense Tracker

Filter

Month: All

All

Category: All

Apply Filter

Add Transaction

Date: 6/8/25

Category: Food

Amount:

Type: Expense

Add

ID	Date	Category	Amount	Type
5	2025-07-01	Salary	1000.0	Income
4	2025-06-15	Transport	8.0	Expense
3	2025-06-08	Food	12.5	Expense
6	2025-06-08	Entertainment	30.0	Expense
8	2025-06-08	Food	1250.5	Income

Remove Selected Transaction

Summary: Expense: \$50.50 | Income: \$2250.50

Income Wate

Progress Bars

Bar Chart

Monthly Chart

Pie Chart

Export to CSV

Toggle Theme

Functional Expense Tracker

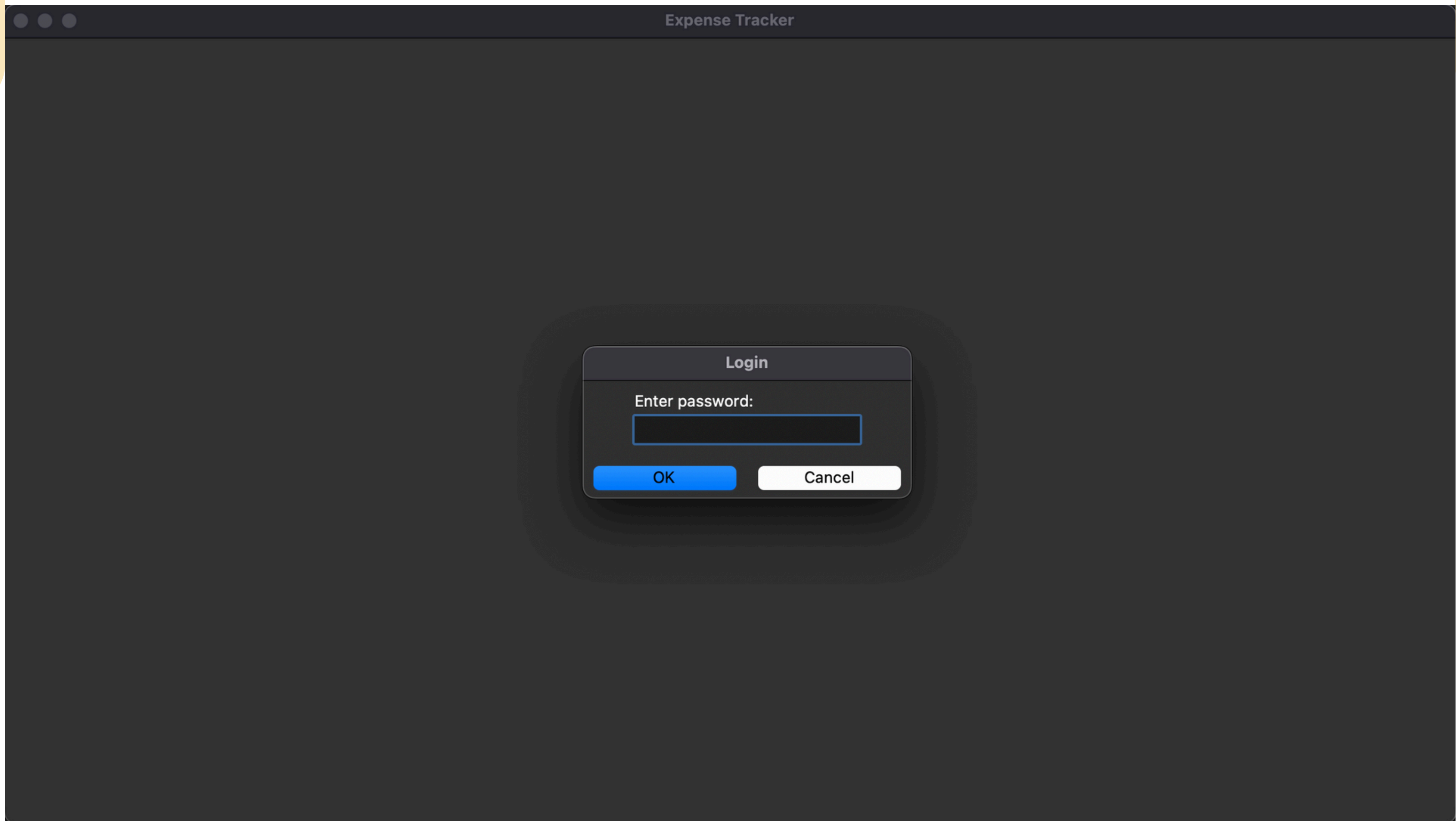
- Allows user to add, remove, and filter transactions.
- Offers category-based filtering and real-time summary updates.
- Since it contains sensitive information, included password function

Data Visualization

- Pie Chart: Shows percentage-wise category expense.
- Bar Chart: Displays actual amount spent per category.
- Waterfall Chart: Visual representation of income vs expenses.
- Monthly Chart: Side-by-side comparison of monthly incomes and expenses.
- Progress Bars: Shows budget consumption per category.



DEMO



Expense Tracker

Login

Enter password:

OK

Cancel

Expense Tracker

Income Waterfall Chart

Filter

Month: All All Category: All Apply Filter

Add Transaction

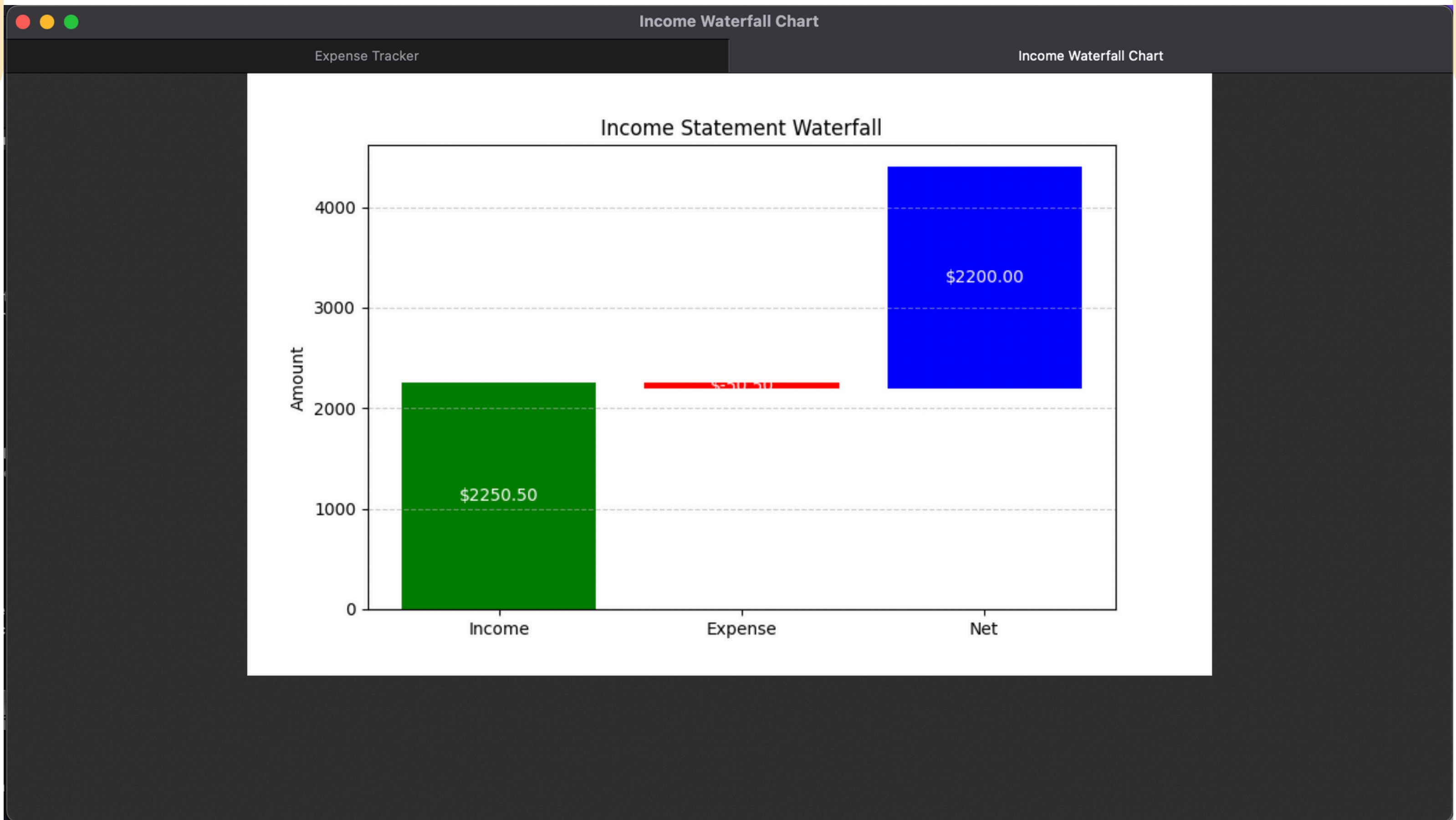
Date: 6/8/25 Category: Food Amount: Type: Expense Add

ID	Date	Category	Amount	Type
5	2025-07-01	Salary	1000.0	Income
4	2025-06-15	Transport	8.0	Expense
3	2025-06-08	Food	12.5	Expense
6	2025-06-08	Entertainment	30.0	Expense
8	2025-06-08	Food	1250.5	Income

Remove Selected Transaction

Summary: Expense: \$50.50 | Income: \$2250.50

Income WateProgress BarsBar ChartMonthly ChartPie ChartExport to CSVToggle Theme



Expense Tracker

Income Waterfall Chart

Filter

Month: All All Category: All Apply Filter

Add Transaction

Date: 6/8/25 Category: Food

Expense

Add

ID	Date	Amount	Type
5	2025-	1000.0	Income
4	2025-	8.0	Expense
3	2025-	12.5	Expense
6	2025-	30.0	Expense
8	2025-	1250.5	Income

Save

Save As:

Tags:

Where: Downloads

Format: CSV files (.csv)

Cancel Save

Remove Selected Transaction

Summary: Expense: \$50.50 | Income: \$2250.50

Income Wate

Progress Bars

Bar Chart

Monthly Chart

Pie Chart

Export to CSV

Toggle Theme

CONCLUSION

This project demonstrates how GUI applications can be built using tkinter combined with powerful libraries like matplotlib for visualization. Modular design, good UI/UX practices, and appropriate data handling make it a robust personal finance tool. The project showcases the integration of database interaction, exception handling, and dynamic plotting to provide meaningful insights to users about their financial behavior



THANK YOU

