

# Personal Finance Manager – Project Documentation

## Project Overview

**Project Name:** Personal Finance Manager

**Description:**

A Python-based command-line application to manage personal finances, including income and expenses. Features include expense tracking, income tracking, category-wise summaries, monthly reports, net savings analysis, backup functionality, and optional data visualization.

**Objectives:**

- Track income and expenses efficiently.
- Generate reports for analysis and decision-making.
- Practice Python OOP, file handling, and data visualization.

---

## Setup Instructions

1. Install **Python 3.x** (preferably 3.12).
2. Clone the repository:

```
git clone https://github.com/Anushage2/personal-finance-manager.git
cd personal-finance-manager
```

3. Install dependencies:

```
pip install -r requirements.txt
```

4. Run the program:

```
python main.py
```

---

## Project Structure

```
personal-finance-manager/
├── main.py           ← Main program entry point
├── expense.py        ← Expense class
├── income.py         ← Income class
├── file_manager.py   ← CSV file operations
├── utils.py          ← Utility functions
├── reports.py        ← Reporting functions
├── data/             ← Sample and user data files
├── backup/           ← Backup data
├── docs/
│   └── project_documentation.docx
```

```
├── user_manual.pdf
└── screenshots/
    ├── menu.png
    ├── add_expense.png
    ├── view_expense.png
    ├── category_summary.png
    ├── monthly_report.png
    ├── add_income.png
    ├── view_income.png
    └── income_vs_expense.png
```

---

## Features

- Add, view, and search expenses.
  - Add and view income.
  - Category-wise expense summary.
  - Monthly report and net savings calculation.
  - Backup and restore functionality.
  - Optional charts for data visualization.
- 

## Code Explanation

- **main.py:** Handles menu and program flow.
  - **expense.py / income.py:** Defines Expense and Income classes.
  - **file\_manager.py:** Reads/writes CSV data.
  - **utils.py:** Validates user input.
  - **reports.py:** Generates summaries and charts.
- 

## Screenshots

**Figure 1: Main Menu**

```
PS C:\Users\safe\Desktop\personal-finance-manager> python main.py

==== PERSONAL FINANCE MANAGER ====
1. Add New Expense
2. View All Expenses
3. Add New Income
4. View All Income
5. View Category-wise Expense Summary
6. Monthly Income Report
7. Income vs Expense Overview (Net Savings)
8. Backup Data
9. Exit
Enter your choice (1-9): █
```

**Figure 2: Add Expense**

```
PS C:\Users\saf\Deskop\personal-finance-manager> python main.py

==== PERSONAL FINANCE MANAGER ====
1. Add New Expense
2. View All Expenses
3. Add New Income
4. View All Income
5. View Category-wise Expense Summary
6. Monthly Income Report
7. Income vs Expense Overview (Net Savings)
8. Backup Data
9. Exit
Enter your choice (1-9): 1
Enter date (YYYY-MM-DD): 2025-11-10
Enter category: shopping
Enter amount: 1000
Note (optional): festival
✓ Expense added successfully!
```

**Figure 3: View All Expenses**

```
Enter your choice (1-9): 2

📅 All Expenses:
```

	Date	Category	Amount	Note
0	2025-01-03	Food	150	Breakfast at cafe
1	2025-01-04	Transport	40	Bus ticket
2	2025-01-05	Groceries	760	Weekly vegetables
3	2025-01-06	Shopping	1200	Clothing items
4	2025-01-08	Entertainment	300	Movie with friends
5	2025-01-09	Food	220	Lunch at restaurant
6	2025-01-12	Health	450	Medicines
7	2025-01-14	Bills	980	Electricity bill
8	2025-01-15	Transport	90	Auto fare
9	2025-01-16	Groceries	1450	Monthly essentials
10	2025-01-18	Shopping	650	Beauty products
11	2025-01-20	Travel	600	Train ticket
12	2025-01-22	Food	180	Snacks and juice
13	2025-01-25	Entertainment	1100	Concert ticket
14	2025-01-28	Bills	300	Water bill
15	2025-02-01	Food	200	Dinner at cafe
16	2025-02-03	Transport	500	Fuel
17	2025-02-05	Shopping	1800	Shoes purchase
18	2025-02-08	Entertainment	850	Theme park entry

**Figure 4: Category-wise Summary**

```
Enter your choice (1-9): 5

📊 Expense Category-wise Summary:
```

category	
Bills	1280.0
Entertainment	2250.0
Food	750.0
Groceries	2210.0
Health	750.0
Shopping	3650.0
Transport	630.0
Travel	600.0
shopping	1000.0

Name: amount, dtype: float64

Figure 5: Monthly Report

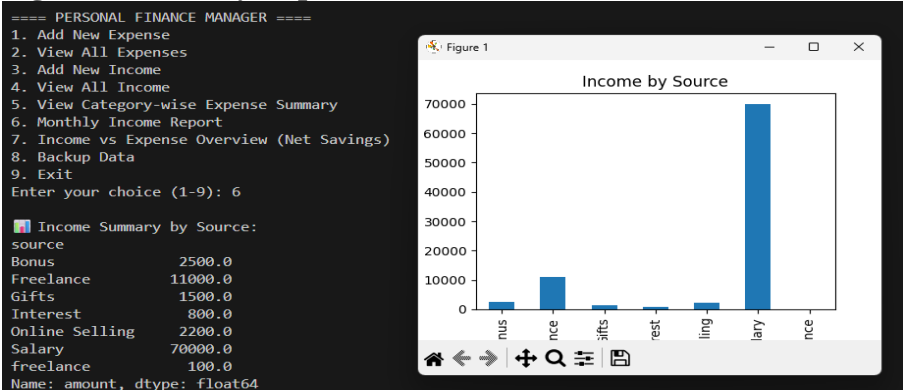


Figure 6: Add Income

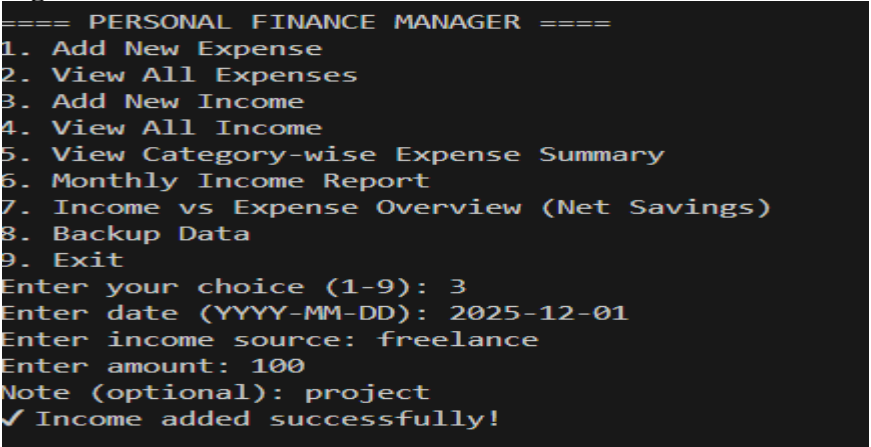
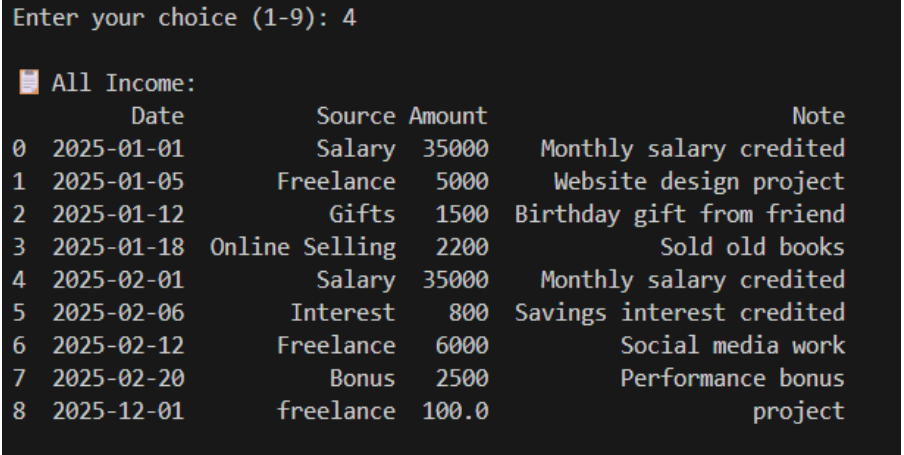
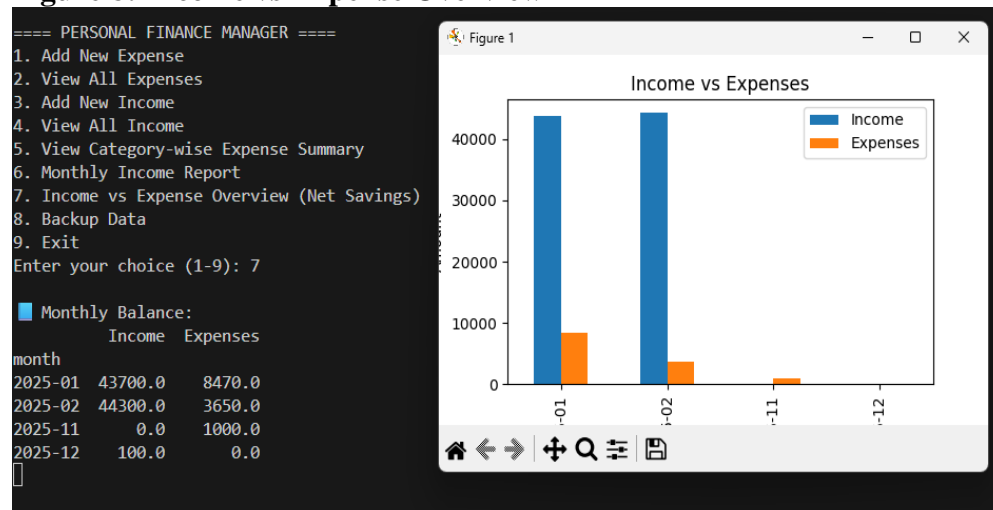


Figure 7: View All Income



**Figure 8: Income vs Expense Overview**



---

## Technical Details

- Language: Python 3.x
- Libraries: pandas, matplotlib, numpy, csv
- Design: OOP (Expense and Income classes)
- File handling: CSV for data persistence
- Validation: User input validation for amount, date, and category

---

## Testing & Validation

- Added multiple expenses and incomes to test CSV read/write.
- Verified reports match input data.
- Tested backup and restore functionality.
- Checked charts display correct data.

---

## Conclusion

This project helps users manage finances effectively and demonstrates Python skills including OOP, file handling, data processing, and visualization.

---