

# **PERSONAL FINANCE MANAGER**

## **User Documentation**

### **1. Introduction**

The Personal Finance Manager is a Python-based console application designed to help users track and manage their daily expenses. The system provides a simple menu-driven interface that allows users to record expenses, view stored data, and generate analytical reports. This document explains the system in a clear and structured manner for academic and user reference.

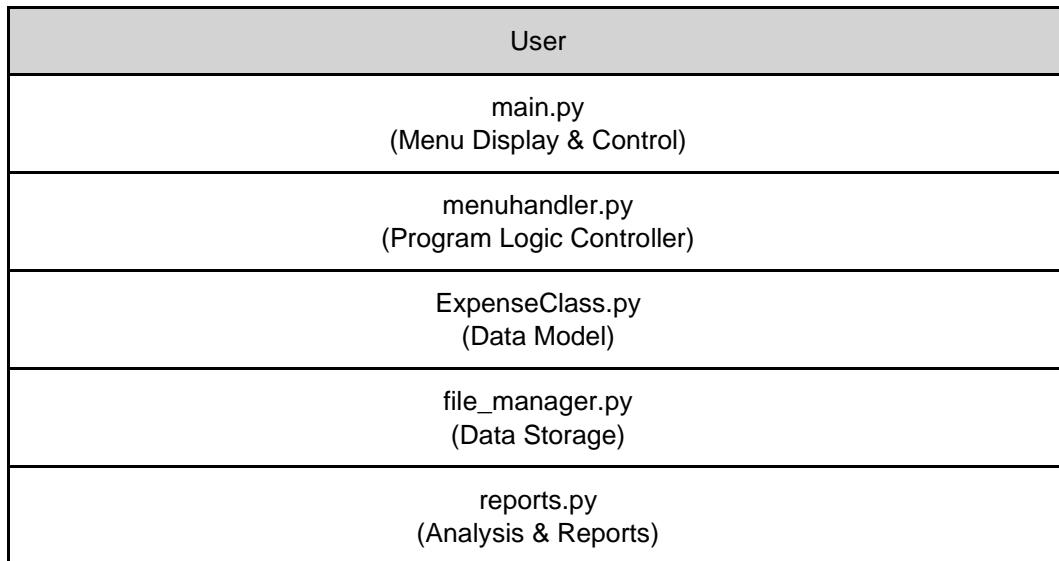
### **2. Objectives**

The main objectives of this application are:

- To maintain a record of daily expenses
- To categorize expenses for better financial understanding
- To generate category-wise and monthly summaries
- To demonstrate modular and object-oriented programming in Python
- To store expense data persistently using CSV files

### 3. System Architecture

The Personal Finance Manager follows a modular architecture where each Python script has a clearly defined responsibility. This design improves readability, maintainability, and scalability of the application.



## 4. Script-wise Explanation

### 4.1 main.py

**Purpose:** Acts as the entry point of the application.

**Why it exists:** To display the main menu and control the program flow.

**Key Functions:**

- `print_menu()`: Displays menu options
- `main()`: Accepts user input and routes execution

### 4.2 menuhandler.py

**Purpose:** Handles user choices and executes corresponding actions.

**Why it exists:** Separates decision-making logic from user interface.

**Key Function:**

- `menuhandler(choice)`: Executes logic based on selected option

### 4.3 ExpenseClass.py

**Purpose:** Defines the structure of an expense.

**Why it exists:** To store expense data in an organized object-oriented manner.

**Key Components:**

- Expense class
- `__str__()`: Formats expense output

### 4.4 file\_manager.py

**Purpose:** Manages persistent storage of expenses.

**Why it exists:** Centralizes CSV file operations.

**Key Functions:**

- `save_expense()`: Saves expense to CSV file
- `read_expenses()`: Reads stored expenses

### 4.5 reports.py

**Purpose:** Generates summaries and reports.

**Why it exists:** Separates analytical logic from data storage.

**Key Functions:**

- `ViewAllExp()`: Displays all expenses
- `CatWiseSummary()`: Category-wise totals
- `MonthlyReport()`: Monthly graphical report

## 5. Step-by-Step Execution Flow

1. User runs the application
2. main.py displays the main menu

```
=====  
PERSONAL FINANCE MANAGER  
=====  
  
MAIN MENU :  
1) Add New Expense  
2) View All Expenses  
3) View Category-wise Summary  
4) Generate Monthly Report  
5) Search Expenses  
6) Backup Data  
7) Exit  
Enter your choice (1-7): |
```

3. User selects a menu option

```
MAIN MENU :  
1) Add New Expense  
2) View All Expenses  
3) View Category-wise Summary  
4) Generate Monthly Report  
5) Search Expenses  
6) Backup Data  
7) Exit  
Enter your choice (1-7): 1  
  
ADD NEW EXPENSE:  
Enter amount: |
```

4. menuhandler.py processes the selection

```
MAIN MENU :  
1) Add New Expense  
2) View All Expenses  
3) View Category-wise Summary  
4) Generate Monthly Report  
5) Search Expenses  
6) Backup Data  
7) Exit  
Enter your choice (1-7): 1  
  
ADD NEW EXPENSE:  
Enter amount: 5000  
Enter category (Food/Transport/Entertainment/Shopping/Other): Food
```

5. Expense data is validated and created
6. Data is saved or analyzed using appropriate modules
7. Output is displayed to the user

```
MAIN MENU :  
1) Add New Expense  
2) View All Expenses  
3) View Category-wise Summary  
4) Generate Monthly Report  
5) Search Expenses  
6) Backup Data  
7) Exit  
Enter your choice (1-7): 1  
  
ADD NEW EXPENSE:  
Enter amount: 5000  
Enter category (Food/Transport/Entertainment/Shopping/Other): Food  
Enter date (YYYY-MM-DD) or press Enter for today: 1999-04-28  
Enter description: Pizza  
1999-04-28 Food 5000.0 Pizza
```

8. Control returns to the main menu

```
MAIN MENU :  
1) Add New Expense  
2) View All Expenses  
3) View Category-wise Summary  
4) Generate Monthly Report  
5) Search Expenses  
6) Backup Data  
7) Exit  
Enter your choice (1-7): 1  
  
ADD NEW EXPENSE:  
Enter amount: 5000  
Enter category (Food/Transport/Entertainment/Shopping/Other): Food  
Enter date (YYYY-MM-DD) or press Enter for today: 1999-04-28  
Enter description: Pizza  
1999-04-28 Food 5000.0 Pizza  
  
Press Enter to return to main menu...  
===== PERSONAL FINANCE MANAGER ======  
  
MAIN MENU :  
1) Add New Expense  
2) View All Expenses  
3) View Category-wise Summary  
4) Generate Monthly Report  
5) Search Expenses  
6) Backup Data  
7) Exit  
Enter your choice (1-7): |
```

## 6. Conclusion

The Personal Finance Manager is a well-structured and user-friendly application that demonstrates essential Python programming concepts such as modular design, object-oriented programming, file handling, and basic data analysis. This documentation provides a clear understanding of the system and is suitable for academic submission and evaluation.