Financial (*Expense*) Management Application

By: Group 20

Members:

Atharv Bisht (25BCE10596)

Aditya Uday Hota (25BCE11176)

Arya Bhagwat (25BAI11273)

Indrajith Sen (25BET10015)

Thiru Selvi (25MEI10025)

Problem Statement & Objective

▶ Managing personal finances is a common challenge faced by individuals.

▶ Our application helps users record their daily income and expenses, categorize them, and analyze their financial balance.

▶ 💡 **Objective**: To promote awareness of spending habits and better savings management.

Project Features

▶ Add, update, and delete income/expense records

▶ Categorize transactions (Food, Travel, Rent, etc.)

▶ View summaries and reports (Income, Expenses, Balance)

▶ Persistent data storage using CSV file

▶ Search by date or category

▶ ✓ **Completed**: Add, View, Report, Storage

▶ ☐ **In Progress**: Update, Delete, Search

Technologies Used

- 📄 Programming Language: Python

- 📁 File Storage: CSV file

- ▫ Interface Type: Console-based Application

Libraries Used:

- csv — for data storage

System Design

Modules:

- **Input Module** — Takes user data (income/expense)
- **Storage Module** — Saves data in CSV
- **Report Module** — Calculates and displays summary
- **UI Module** — Console-based user interface

□ **Flow**:

- [User] → [Input Module] → [CSV Storage] → [Report Module] → [Display Output]

Sample Input

```
1. Add record
2. View records
3. Report
4. Exit
Enter choice: 1
Enter date (DD-MM-YYYY): 28-10-2025
Enter type(income/expense): income
Enter category: food
Enter amount: 5000
Record added successfully...

1. Add record
2. View records
3. Report
4. Exit
Enter choice: 1
Enter date (DD-MM-YYYY): 29-10-2025
Enter type(income/expense): expense
Enter category: travel
Enter amount: 600
Record added successfully...
```

Here, we have taken two inputs.

▶ The first input is "income" and the second is "expense"

View records and report

View Records:

```
1. Add record
2. View records
3. Report
4. Exit
Enter choice: 2
['Date', 'Type', 'Category', 'Amount']
['28-10-2025', 'income', 'food', '5000']
['29-10-2025', 'expense', 'travel', '600']
```

View Report:

```
1. Add record
2. View records
3. Report
4. Exit
Enter choice: 3
......REPORT......
Total income:  5000
Total expense:   600
Balance:  4400
```

Data stored in csv file

	A	B	C	D
1	Date	Type	Category	Amount
2	28-10-2025	income	food	5000
3	29-10-2025	expense	travel	600

finance

A csv file named "*finance*" will be saved on the device which stores the entered data.

Current Progress

✓ **Code Modules Completed**: Add, View, Report

✓ **Data Storage**: Implemented via CSV

✓ **UI**: Console-based interactive menu

✓ **Working Demo**: Fully functional

Next Steps:

▶ Implement update/delete functionality

THANK YOU