"SAFE MONEY BANKING SYSTEM"

By: Sujeeth and Yaswanth
Group: Slytherins

[IITG]

1 Introduction

The "SafeMoney Banking" application is a Python-based banking system developed as a course project by Sujeeth and Yaswanth of the Slytherins group. This project simulates a real-world banking environment, offering users a secure platform to manage finances with features like account creation, deposits, withdrawals, fixed deposits (FDs), transaction history, and spending visualization. It uses Python's object-oriented programming, a Tkinter GUI, and CSV files for data persistence.

1.1 Objectives

- Design a secure banking system with PIN authentication.
- Implement deposits, withdrawals, and FDs.
- Visualize spending patterns graphically.
- Ensure data persistence via files.

1.2 Technologies Used

- **Python**: Core logic.
- **Tkinter**: GUI framework.
- **Pandas & NumPy**: Data handling.
- **Matplotlib**: Graphing.
- **Pytz**: Time zones.
- **CSV**: Storage.

2 Implementation

The application is built around two classes:

2.1 Customer Class

Manages banking operations:

- **Account Creation**: Generates 4-digit IDs, validates inputs, stores in `userdet.csv`.
- **Transactions**: Updates `userdet.csv`, logs in `t.csv` with IST timestamps.
- **FDs**: Offers 6% or 7.5% interest based on age, stored in `fd.csv`.
- **Visualization**: Plots balance trends with Matplotlib.

2.2 BankingApp Class

Provides the GUI:

- **Main Menu**: Create account, login, exit.
- **Logged-in Menu**: Access to all features.

3 Results

The application was tested successfully:

- **Account Management**: Secure creation and updates.
- **Transactions**: Accurate updates and logging.
- **FDs**: Correct interest calculations.
- **Visualization**: Effective trend graphs.

Testing confirmed data consistency and error handling (e.g., invalid PINs), meeting all objectives.

3.1 Sample Code

```
def self_deposit(self, amount):
    try:
        amount = float(amount)
        if amount <= 0:
            return False, "Amount must be positive"
        ist = pytz.timezone('Asia/Kolkata')
        now_ist = datetime.now(ist)
        formatted_now = now_ist.strftime("%Y-%m-%d %H:%M:%S")
        with open("userdet.csv", "r", newline="") as f2:
            r1 = list(csv.reader(f2))
        for row in r1:
            if row and int(row[0]) == self.account_number:
                row[3] = float(row[3]) + amount
                self.balance = float(row[3])
                break
```

```
16          with open("userdet.csv", "w", newline="") as f1:
17              w1 = csv.writer(f1)
18              w1.writerows(r1)
19          with open("t.csv", "a", newline="") as f1:
20              w1 = csv.writer(f1)
21              w1.writerow([self.account_number, formatted_now,
                    amount, self.balance])
22              self.transactionhistory.append((self.account_number,
                    formatted_now, amount, self.balance))
23          return True, f"Deposit successful. New balance:
                {self.balance}"
24      except Exception as e:
25          return False, f"Error: {str(e)}"
```

Listing 1: Deposit Method

4 Contributions

4.1 Sujeeth

- Implemented `Customer` class (accounts, transactions, FDs).
- Managed CSV operations and testing.

4.2 Yaswanth

- Developed `BankingApp` class and GUI.
- Integrated Matplotlib and documented the project.

Both collaborated on integration and testing.

5 Conclusion

"SafeMoney Banking" by Sujeeth and Yaswanth (Slytherins) delivers a functional banking system, showcasing Python's capabilities. Future enhancements could include databases and encryption. This project enhanced our programming and teamwork skills.