**Report for project.**

**Project Title: Bank Management System**

**Contents:**

* Introduction
* Objectives
* Functional Overview
* File Descriptions
* Code Structure
* Features and Functions
* Usage Instructions
* Future Enhancements

**Introduction**

This project is a Bank Management System that simulates a banking environment for customer account creation, money transfer, loan detailing, fixed deposit management, and more. It enables users to manage accounts and perform banking operations such as depositing money, transferring funds, checking interest on fixed deposits, and calculating the future value of money under inflation.

**Objectives**

* Implement a bank management system to manage customer accounts.
* Provide functions for account creation, money transfer, interest calculation, and fixed deposits.
* Offer user-friendly options for banking operations through a menu-driven interface.

**Functional Overview**

**The project allows the user to:**

* Create new accounts with customer details (Aadhar, PAN, etc.).
* Display account details for all accounts or a specific account.
* Transfer money between accounts.
* Calculate cost/value of money after N years considering inflation.
* Display EMI details for loans.
* Calculate Fixed Deposit interest for specified amounts and durations.

**File Descriptions**

**BankProject.c**: Main source file containing all the functions and logic for bank operations.

**log.txt**: A file where customer account details are logged after account creation.

**Amount.txt**: A file where the account numbers and their respective balances are recorded.

**Code Structure**

The project consists of two key structures:

* BankAccount: Stores individual customer information such as name, age, Aadhar, PAN, account number, balance, etc.
* Account: Used for basic account information, including name, account number, and balance for transactions and display purposes.

The project is divided into multiple functions for ease of modularity and code readability:

1. **readAccounts()**: Reads account details from the file.
2. **findAccount()**: Locates an account using the account number.
3. **transferAccount()**: Handles money transfers between accounts.
4. **createAccount()**: Allows users to create a new account with validations.
5. **costOfMoney()**: Calculates the future cost of money.
6. **valueOfMoney()**: Calculates the future value of money considering inflation.
7. **FixedDepositInterest()**: Computes interest for a fixed deposit.
8. **principal()**: Displays the loan amortization schedule based on EMI calculations

**Features and Functions**

**1. Account Creation**

* **Input**: User provides Aadhar, PAN, first/last name, address, age, and deposit amount.
* **Validations**: Aadhar and PAN formats are validated. Age must be above 17.
* **Files**: Information is stored in log.txt and Amount.txt.

**2. Display Accounts**

* Displays all accounts and allows for searching by account number.

**3. Money Transfer**

* Facilitates the transfer of money between two valid accounts. Checks are made for sufficient funds in the sender’s account.

**4. Value and Cost of Money**

* **Cost of Money**: Computes how much a sum of money will be worth after N years considering an inflation rate.
* **Value of Money**: Computes the depreciation of money over time due to inflation.

**5. Loan EMI Calculation**

* **EMI Calculation**: Calculates monthly EMI based on interest rate and loan tenure.
* Displays the remaining principal after each payment cycle.

**6. Fixed Deposit Interest Calculation**

* Calculates the maturity amount for a fixed deposit over N years at a specified interest rate.

**7. Exiting the Program**

* Safely exits the program and saves data.

**Usage Instructions**

**1.Compiling the Project**

**2.Running the Program:**

* After compilation, run the program

**3.Menu Options:**

On running the program, a menu will be displayed with the following options:

* + - Create an Account
    - Display All Accounts
    - Display Single Account
    - Transfer Money
    - Calculate Cost of Money After N Years
    - Calculate Value of Money After N Years
    - Loan Detailing
    - Fixed Deposit Interest
    - Exit

**Future Enhancements**

**Graphical User Interface (GUI):**

Transition from a text-based system to a graphical interface using libraries like GTK or Qt for better user interaction.

**Authentication System:**

Implement a user login system with account verification.

**Database Integration:**

Replace text files with a database like SQLite or MySQL for better data management.

**Mobile Integration:**

Develop a mobile application using React Native or Flutter for mobile banking.

**Conclusion**

This project is a comprehensive implementation of a basic bank management system. It provides essential banking operations like account management, money transfer, and various financial calculations. With the modular code structure, it can be easily enhanced and expanded to include additional features like loans, interest calculation, and reporting.