✅ Bank Management System – Python Functions Only

---

1. Problem Statement

Simulate a **Bank Management System** in Python with **Admin** and **Customer** perspectives:

- **Admin**: Can view all accounts, approve/decline loan requests, manage accounts.
- **Customer**: Can view balance, deposit, withdraw, transfer money, and apply for loans.

---

2. Features & Functionalities

Admin Features

1. **View All Accounts** – List all customer accounts with details.
2. **View Loan Requests** – See pending loan requests.
3. **Approve/Reject Loan Requests** – Update status for each request.
4. **Exit** – Close Admin menu.

Customer Features

1. **Check Balance** – View own account balance.
2. **Deposit Money** – Add money to account.
3. **Withdraw Money** – Withdraw money if sufficient balance.
4. **Transfer Money** – Send money to another account.
5. **Apply for Loan** – Submit a loan request.
6. **View Loan Status** – Check if loan is approved/rejected.
7. **Exit** – Close Customer menu.

---

3. Requirements

- Python 3.8+
- Knowledge of:
    - Lists & dictionaries
    - Functions
    - Loops & conditionals

---

4. Data Structures

Customers List

```
customers = [
    {"id": 1, "name": "Ravi", "balance": 5000, "pin": 1234, "loan_requests": []},
    {"id": 2, "name": "Priya", "balance": 7000, "pin": 2345, "loan_requests": []},
    {"id": 3, "name": "Suresh", "balance": 10000, "pin": 3456, "loan_requests": []}
]
```

Loan Requests List

```
loan_requests = [
    {"customer_id": 1, "amount": 5000, "status": "Pending"},
    {"customer_id": 2, "amount": 10000, "status": "Pending"}
]
```

---

5. Functions to Implement

Admin Functions

- view_all_accounts(customers) – Display all customer accounts.
- view_loan_requests(loan_requests, customers) – Display pending loan requests.
- approve_loan(loan_requests, customer_id) – Approve a loan request.
- reject_loan(loan_requests, customer_id) – Reject a loan request.

Customer Functions

- check_balance(customers, name, pin) – Show balance if PIN matches.
- deposit(customers, name, pin, amount) – Deposit money.
- withdraw(customers, name, pin, amount) – Withdraw money.
- transfer_money(customers, name, pin, receiver_name, amount) – Transfer to another customer.
- apply_loan(loan_requests, customers, name, amount) – Submit loan request.
- view_loan_status(loan_requests, customers, name) – Check status of loan requests.

Menu Functions

- admin_menu(customers, loan_requests) – Display admin menu & handle input.
- customer_menu(customers, loan_requests) – Display customer menu & handle input.

---

6. Menu Example

Admin Menu

===== Admin Menu =====

1. View All Accounts

2. View Loan Requests

3. Approve Loan

4. Reject Loan

5. Exit

Enter your choice:

Customer Menu

===== Customer Menu =====

1. Check Balance

2. Deposit Money

3. Withdraw Money

4. Transfer Money

5. Apply for Loan

6. View Loan Status

7. Exit

Enter your choice:

---

7. Step-by-Step To-Do Checklist

Step 1 – Setup

- Create bank_system.py
- Define predefined customers and empty loan_requests.
- Define empty functions with pass.
- Create main menu to select **Admin** or **Customer** view.

Step 2 – Admin Functions

- View all accounts → list all customers with balance.
- View pending loan requests.
- Approve/Reject loan → update status.

Step 3 – Customer Functions

- Verify **name + PIN** before any operation.
- Check balance, deposit, withdraw.

- Transfer money → subtract from sender, add to receiver.
- Apply for loan → add to loan_requests with status "Pending".
- View loan status → show approved/rejected/pending loans.

Step 4 – Exit Options

- Return to main menu or exit program.

Step 5 – Testing

- Test all **Admin operations**.
- Test all **Customer operations**.
- Test PIN verification.
- Test loan approval workflow.
- Test money transfers.

---

8. Evaluation Criteria

- Correct implementation of **Admin & Customer operations** – 50%
- Proper use of functions & validation – 20%
- Correct handling of balances, transfers, and loan statuses – 15%
- Readable, commented code – 15%