✅ ATM Machine Project (Python Functions Only – Name & PIN Authentication)

---

1. Problem Statement

Simulate a simple **ATM Machine** using Python for **predefined 10 accounts**.
Users can **check balance, deposit, withdraw, and view transaction history** using **name and PIN** for authentication.

---

2. Features & Functionalities

1. **Check Balance** – View current balance for an account.
2. **Deposit Money** – Add money to an account.
3. **Withdraw Money** – Withdraw money if balance is sufficient.
4. **Transaction History** – View all deposits and withdrawals.
5. **Exit Program** – Close the application.

---

3. Requirements

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

---

4. Predefined Accounts Data (10 Accounts)

Each account has: name, balance, pin, transactions (list).

```
accounts = [
  {"name": "Ravi", "balance": 5000, "pin": 1234, "transactions": []},
  {"name": "Priya", "balance": 7000, "pin": 2345, "transactions": []},
  {"name": "Suresh", "balance": 10000, "pin": 3456, "transactions": []},
  {"name": "Anita", "balance": 3000, "pin": 4567, "transactions": []},
  {"name": "Kiran", "balance": 8500, "pin": 5678, "transactions": []},
  {"name": "Meena", "balance": 12000, "pin": 6789, "transactions": []},
  {"name": "Vamsi", "balance": 4500, "pin": 7890, "transactions": []},
  {"name": "Rohit", "balance": 6000, "pin": 8901, "transactions": []},
```

{"name": "Sonia", "balance": 9000, "pin": 9012, "transactions": []},

{"name": "Neha", "balance": 11000, "pin": 1122, "transactions": []}

]

---

5. Functions to Implement (Arguments: name, pin)

- check_balance(accounts, name, pin) – Show balance if name & PIN match.
- deposit(accounts, name, pin, amount) – Add money & record transaction.
- withdraw(accounts, name, pin, amount) – Withdraw money if sufficient balance & record transaction.
- transaction_history(accounts, name, pin) – Show all deposits/withdrawals.
- menu() – Display menu & handle user input (ask for name & PIN per operation).

---

6. Menu Example

===== ATM Machine =====

1. Check Balance

2. Deposit Money

3. Withdraw Money

4. Transaction History

5. Exit

Enter your choice:

---

7. Step-by-Step To-Do Checklist

Step 1 – Setup

- Create atm_machine.py
- Copy the 10 predefined accounts into the program.
- Define empty functions (pass) with proper arguments: name, pin, (and amount for deposit/withdraw).
- Prepare while loop for menu options.

Step 2 – Check Balance

- Ask for **Name** and **PIN**.
- Pass name and pin to check_balance().
- If match → print balance.

- Else → print “Invalid name or PIN”.

Step 3 – Deposit Money

- Ask for **Name**, **PIN**, and **Deposit Amount**.
- Pass to deposit().
- Update balance & add "Deposited ₹amount" to transactions.
- Confirm success.

Step 4 – Withdraw Money

- Ask for **Name**, **PIN**, and **Withdraw Amount**.
- Pass to withdraw().
- Check sufficient balance.
- Update balance & add "Withdrew ₹amount" to transactions.
- Confirm success or show “Insufficient balance”.

Step 5 – Transaction History

- Ask for **Name** and **PIN**.
- Pass to transaction_history().
- Display all deposits & withdrawals.

Step 6 – Exit

- Break loop when user chooses Exit.

Step 7 – Testing

- Test **Check Balance, Deposit, Withdraw, Transaction History** for multiple accounts.
- Enter **wrong PIN** → should deny access.
- Verify correct transactions are recorded.

---

8. Evaluation Criteria

- Correct banking operations with **PIN verification** – 50%
- Proper use of functions with arguments – 20%
- Validation of balance & correct transaction records – 15%
- Readable code with comments – 15%