```python
print("PROJECT BY - ")
print("              |AASHISH VICTOR|")
import json, os
from datetime import datetime

DB_FILE = "atm_db.json"

def load_db():
    if not os.path.exists(DB_FILE):
        return {"accounts": {}, "next_ac_no": 10001}
    try:
        with open(DB_FILE, "r", encoding="utf-8") as f:
            return json.load(f)
    except (json.JSONDecodeError, OSError):
        return {"accounts": {}, "next_ac_no": 10001}

def save_db(db):
    with open(DB_FILE, "w", encoding="utf-8") as f:
        json.dump(db, f, indent=2)

def input_pin(prompt="Enter 4-digit PIN: "):
    while True:
        pin = input(prompt).strip()
        if pin.isdigit() and len(pin) == 4:
            return pin
        print("Invalid PIN. Please enter exactly 4 digits.")

def input_amount(prompt="Enter amount: "):
    while True:
        txt = input(prompt).strip()
        if txt.isdigit() and int(txt) > 0:
            return int(txt)
        print("Invalid amount. Enter a positive integer (e.g., 500).")

def create_account(db):
    print("\n--- Create New Account ---")
    name = input("Enter full name: ").strip()
```

```python
def create_account(db):
    print("\n--- Create New Account ---")
    name = input("Enter full name: ").strip()
    if not name:
        print("Name cannot be empty.")
        return
    pin = input_pin("Set a 4-digit PIN: ")
    ac_no = str(db["next_ac_no"])
    db["next_ac_no"] += 1
    db["accounts"][ac_no] = {"name": name, "pin": pin, "balance": 0, "transactions": []}
    save_db(db)
    print(f"Account created successfully!\nYour Account Number is: {ac_no}")

def login(db):
    print("\n--- Login ---")
    ac_no = input("Enter Account Number: ").strip()
    if ac_no not in db["accounts"]:
        print("Account not found.")
        return None
    pin = input_pin("Enter PIN: ")
    if db["accounts"][ac_no]["pin"] != pin:
        print("Incorrect PIN.")
        return None
    print(f"Welcome, {db['accounts'][ac_no]['name']}!")
    return ac_no

def deposit(db, ac_no):
    print("\n--- Deposit ---")
    amt = input_amount("Enter deposit amount: ")
    db["accounts"][ac_no]["balance"] += amt
    db["accounts"][ac_no]["transactions"].append(
        {"type": "DEPOSIT", "amount": amt, "time": datetime.now().isoformat(timespec="seconds")}
    )
    save_db(db)
    print(f"₹{amt} deposited successfully. New Balance: ₹{db['accounts'][ac_no]['balance']}")

def withdraw(db, ac_no):
    print("\n--- Withdraw ---")
    amt = input_amount("Enter withdrawal amount: ")
    bal = db["accounts"][ac_no]["balance"]
    if amt > bal:
        print("Insufficient balance.")
        return
    db["accounts"][ac_no]["balance"] -= amt
    db["accounts"][ac_no]["transactions"].append(
        {"type": "WITHDRAW", "amount": amt, "time": datetime.now().isoformat(timespec="seconds")}
    )
    save_db(db)
    print(f"₹{amt} withdrawn successfully. New Balance: ₹{db['accounts'][ac_no]['balance']}")

def show_balance(db, ac_no):
    print(f"\n--- Balance ---\nCurrent Balance: ₹{db['accounts'][ac_no]['balance']}")

def mini_statement(db, ac_no, limit=10):
```

```python
def show_balance(db, ac_no):
    print(f"\n--- Balance ---\nCurrent Balance: ₹{db['accounts'][ac_no]['balance']}")

def mini_statement(db, ac_no, limit=10):
    print("\n--- Mini Statement (Last 10) ---")
    txns = db["accounts"][ac_no]["transactions"][-limit:]
    if not txns:
        print("No transactions yet.")
        return
    for t in txns:
        print(f"{t['time']} | {t['type']:8s} | ₹{t['amount']}")
    print(f"Current Balance: ₹{db['accounts'][ac_no]['balance']}")

def change_pin(db, ac_no):
    print("\n--- Change PIN ---")
    old = input_pin("Enter current PIN: ")
    if db["accounts"][ac_no]["pin"] != old:
        print("Incorrect current PIN.")
        return
    new = input_pin("Enter new 4-digit PIN: ")
    if new == old:
        print("New PIN cannot be the same as the old PIN.")
        return
    db["accounts"][ac_no]["pin"] = new
    save_db(db)
    print("PIN changed successfully.")

def user_menu(db, ac_no):
    while True:
        print("\n=== ATM - User Menu ===")
        print("1. Deposit")
        print("2. Withdraw")
        print("3. Balance Enquiry")
        print("4. Mini Statement")
        print("5. Change PIN")
        print("6. Logout")
        choice = input("Select option (1-6): ").strip()
        if choice == "1":
            deposit(db, ac_no)
        elif choice == "2":
            withdraw(db, ac_no)
        elif choice == "3":
            show_balance(db, ac_no)
        elif choice == "4":
            mini_statement(db, ac_no)
        elif choice == "5":
            change_pin(db, ac_no)
        elif choice == "6":
            print("Logged out.\n")
            break
        else:
            print("Invalid choice. Please select 1-6.")

def main_menu():
```

```
                print("Invalid choice. Please select 1-6.")

def main_menu():
    db = load_db()
    print("                                  ")
    print("                        ============")
    print("        ATM SIMULATION           ")
    print("=========                        ")
    while True:
        print("\n--- Main Menu ---")
        print("1. Create New Account")
        print("2. Login")
        print("3. Exit")
        choice = input("Select option (1-3): ").strip()
        if choice == "1":
            create_account(db)
        elif choice == "2":
            ac_no = login(db)
            if ac_no:
                user_menu(db, ac_no)
        elif choice == "3":
            print("Thank you for using the ATM. Goodbye!")
            break
        else:
            print("Invalid choice. Please select 1-3.")

if __name__ == "__main__":
    main_menu()
```