BANK MANAGEMENT SYSTEM PROJECT GUIDELINES

Objective:

The objective of this project is to implement a Bank Management System using Object-Oriented Programming principles, focusing on Inheritance and Abstraction concepts. The system will handle both Conventional and Sharia-compliant Savings Account types, include a login mechanism for account holders, and provide the functionality to export transaction history.

Project Guidelines:

1. Create an abstract class **SavingsAccount** that will serve as the base class for both Conventional and Sharia-compliant Savings Account classes.

   The **SavingsAccount** class should have the following attributes:
   - account_number: A unique identifier for the account.
   - account_holder: The name of the account holder.
   - balance: The current account balance.
   - interest_rate: The annual interest rate for the account.
   - transaction_history: A list to store the transaction history for the account.

   The **SavingsAccount** class should have the following methods:
   - deposit(amount): Add the given amount to the account balance and record the transaction in the history.
   - withdraw(amount): Deduct the given amount from the account balance if sufficient funds are available and record the transaction in the history.
   - calculate_interest(): Calculate and return the interest earned based on the current balance and interest rate.
   - display_account_info(): Display the account information, including the account number, account holder's name, and current balance.

2. Create two classes that inherit from **SavingsAccount**: **ConventionalSavings** and **ShariaSavings**.

   The **ConventionalSavings** class should have an additional attribute:
   - min_balance: The minimum balance required to keep the account active. If the balance goes below this limit, a penalty will be applied.

   The **ShariaSavings** class should override the calculate_interest() method to implement Sharia-compliant interest calculation. (interest_rate = 0%)

3. Implement appropriate constructors for all classes to initialize the attributes.
   Write a **Bank** class that will manage multiple accounts. The **Bank** class should have the following methods:
   - register_account(account): Register a new account in the bank.
   - login(account_number): Verify the account number for login and return the corresponding account object.
   - export_transaction_history(account): Export the transaction history for a given account to local .xlsx and a SPREADSHEET file.
     Day2: OOP - Google Sheets

   Write a program to demonstrate the functionality of the Bank Management System:
   - Create instances of both ConventionalSavings and ShariaSavings accounts.
   - Register these accounts with the Bank.
   - Implement a login mechanism for account holders to access their respective accounts. (just simple login, user: S0001, pass: 12345. No need for hashing)
   - Prefix C for Conventional Account Prefix S for Sharia
   - Perform deposits and withdrawals on the logged-in accounts.
   - Display the account information and the interest earned for both accounts.
   - Run the app and user interaction in terminal.

Another Project Guidelines:

- Use proper naming conventions for variables, functions, and classes.
- Apply appropriate access modifiers for class attributes and methods.
- Handle edge cases, such as insufficient balance during withdrawals or invalid inputs.
- Make use of inheritance, abstraction, and override principles effectively.
- Utilize CSV file handling, pandas, gspread etc for transaction history export.