

Dear Students,

We have covered core python, now it's time to test your coding and logical skills, also to know how confident you are with the syntax, i'm giving you a small project:

Write a python program to replicate a Banking system. The following features are mandatory:

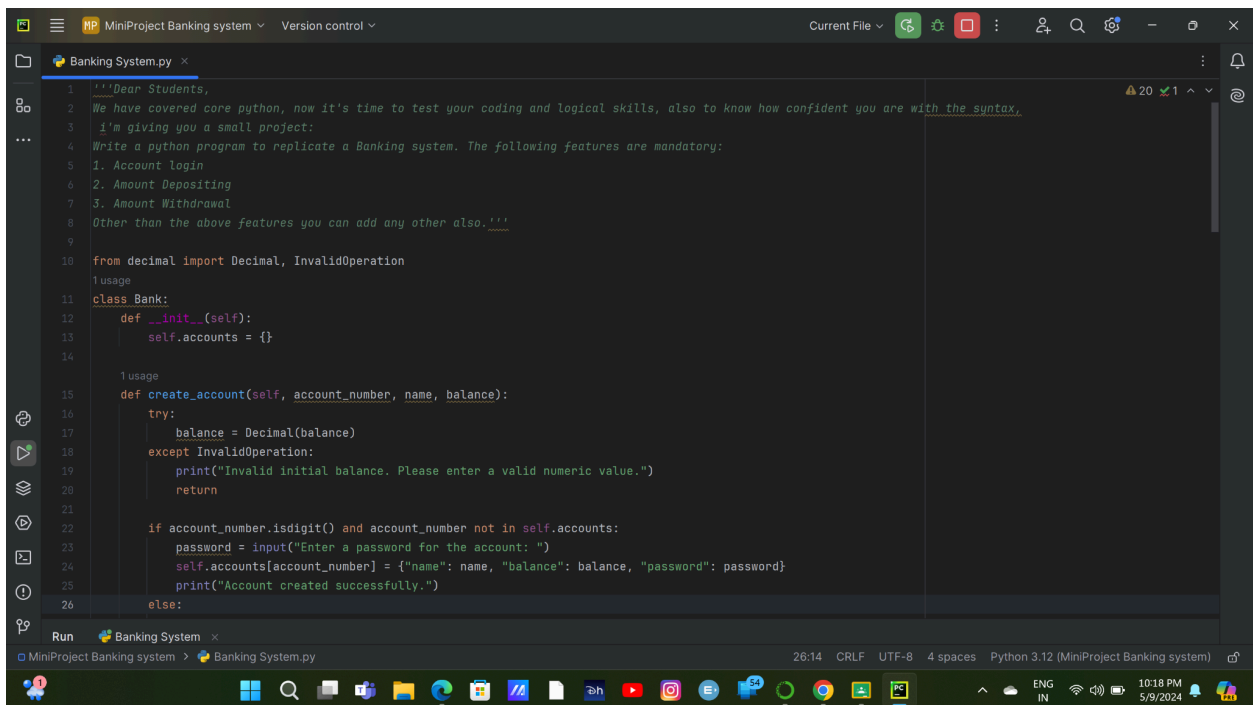
1. Account login
2. Amount Depositing
3. Amount Withdrawal

Other than the above features you can add any other also.

You should upload the code and output in to your github account. Submit the github link here.

(Score : code (7) + extra features (2) + Timely submission (1))

## OMINI PROJECT-BANKING SYSTEM



```
1 '''Dear Students,  
2 We have covered core python, now it's time to test your coding and logical skills, also to know how confident you are with the syntax,  
3 i'm giving you a small project:  
4 Write a python program to replicate a Banking system. The following features are mandatory:  
5 1. Account login  
6 2. Amount Depositing  
7 3. Amount Withdrawal  
8 Other than the above features you can add any other also.'''  
9  
10 from decimal import Decimal, InvalidOperation  
11 usage  
12 class Bank:  
13     def __init__(self):  
14         self.accounts = {}  
15  
16     usage  
17     def create_account(self, account_number, name, balance):  
18         try:  
19             balance = Decimal(balance)  
20         except InvalidOperation:  
21             print("Invalid initial balance. Please enter a valid numeric value.")  
22             return  
23  
24         if account_number.isdigit() and account_number not in self.accounts:  
25             password = input("Enter a password for the account: ")  
26             self.accounts[account_number] = {"name": name, "balance": balance, "password": password}  
27             print("Account created successfully.")  
28         else:  
29             print("Account already exists.")
```

```
MiniProject Banking system  Version control  Current File  20 1 ^ v
Banking System.py x
11 class Bank:
12     def create_account(self, account_number, name, balance):
13         if account_number.isdigit() and account_number not in self.accounts:
14             password = input("Enter a password for the account: ")
15             self.accounts[account_number] = {'name': name, 'balance': balance, 'password': password}
16             print("Account created successfully.")
17         else:
18             print("Invalid account number or account already exists.")
19
20     1 usage
21     def login(self, account_number, password):
22         if account_number in self.accounts and self.accounts[account_number]['password'] == password:
23             print(f"Logged in as {self.accounts[account_number]['name']}")
24             return True
25         else:
26             print("Invalid account number or password.")
27             return False
28
29     1 usage
30     def deposit(self, account_number, amount):
31         if account_number in self.accounts:
32             try:
33                 amount = Decimal(amount)
34             except InvalidOperation:
35                 print("Invalid deposit amount. Please enter a valid numeric value.")
36                 return
37             if amount > 0:
```

```
MiniProject Banking system  Version control  Current File  26:14 CRLF UTF-8 4 spaces Python 3.12 (MiniProject Banking system)
Banking System.py x
11 class Bank:
37     def deposit(self, account_number, amount):
44         if amount > 0:
45             self.accounts[account_number]['balance'] += amount
46             print(f"Deposited ${amount:.2f}. New balance: ${self.accounts[account_number]['balance']:.2f}")
47         else:
48             print("Invalid deposit amount.")
49         else:
50             print("Account does not exist.")
51
52     1 usage
53     def withdraw(self, account_number, password, amount):
54         if account_number in self.accounts and self.accounts[account_number]['password'] == password:
55             try:
56                 amount = Decimal(amount)
57             except InvalidOperation:
58                 print("Invalid withdrawal amount. Please enter a valid numeric value.")
59                 return
60
61             if amount > 0 and self.accounts[account_number]['balance'] >= amount:
62                 self.accounts[account_number]['balance'] -= amount
63                 print(f"Withdrew ${amount:.2f}. New balance: ${self.accounts[account_number]['balance']:.2f}")
64             elif amount <= 0:
65                 print("Invalid withdrawal amount.")
66             else:
67                 print("Insufficient balance.")
68         else:
```

```
MiniProject Banking system  Version control  Current File  20 1 20 10:18 PM 5/9/2024

Banking System.py x
11 class Bank:
53     def withdraw(self, account_number, password, amount):
67         print("Insufficient balance.")
68     else:
69         print("Account does not exist or incorrect password.")
70
71
72 print("$$$WELCOME TO INDIA BANK$$$")
73
74 bank = Bank()
75
76 while True:
77     print("\n1. Create Account\n2. Login\n3. Deposit\n4. Withdraw\n5. Quit")
78     choice = input("Enter your choice: ")
79
80     if choice == "1":
81         account_number = input("Enter account number: ")
82         name = input("Enter your name: ")
83         balance = input("Enter initial balance: ")
84         bank.create_account(account_number, name, balance)
85     elif choice == "2":
86         account_number = input("Enter account number: ")
87         password = input("Enter your password: ")
88         bank.login(account_number, password)
89     elif choice == "3":
90         account_number = input("Enter account number: ")
91         amount = input("Enter the amount to deposit: ")
92         bank.deposit(account_number, amount)

Run  Banking System x
MiniProject Banking system > Banking System.py  26:14 CRLF UTF-8 4 spaces Python 3.12 (MiniProject Banking system)
```

```
MiniProject Banking system  Version control  Current File  72:37 CRLF UTF-8 4 spaces Python 3.12 (MiniProject Banking system) 10:19 PM 5/9/2024

Banking System.py x
75
76 while True:
77     print("\n1. Create Account\n2. Login\n3. Deposit\n4. Withdraw\n5. Quit")
78     choice = input("Enter your choice: ")
79
80     if choice == "1":
81         account_number = input("Enter account number: ")
82         name = input("Enter your name: ")
83         balance = input("Enter initial balance: ")
84         bank.create_account(account_number, name, balance)
85     elif choice == "2":
86         account_number = input("Enter account number: ")
87         password = input("Enter your password: ")
88         bank.login(account_number, password)
89     elif choice == "3":
90         account_number = input("Enter account number: ")
91         amount = input("Enter the amount to deposit: ")
92         bank.deposit(account_number, amount)
93     elif choice == "4":
94         account_number = input("Enter account number: ")
95         password = input("Enter your password: ")
96         amount = input("Enter the amount to withdraw: ")
97         bank.withdraw(account_number, password, amount)
98     elif choice == "5":
99         break
100     else:
101         print("Invalid choice. Please try again.")

Run  Banking System x
MiniProject Banking system > Banking System.py  72:37 CRLF UTF-8 4 spaces Python 3.12 (MiniProject Banking system)
```

```
MiniProject Banking system  Version control  Current File  Run  Banking System.py  Banking System.py  Run  Banking System.py  "C:\Users\sarat\PycharmProjects\MiniProject\MiniProject Banking system\.venv\Scripts\python.exe" "C:\Users\sarat\PycharmProjects\MiniProject\MiniProject Banking sys  
$$$WELCOME TO INDIA BANK$$$  
  
1. Create Account  
2. Login  
3. Deposit  
4. Withdraw  
5. Quit  
Enter your choice: 1  
Enter account number: 14785  
Enter your name: SARATH  
Enter initial balance: 1000  
Enter a password for the account: SARATH07  
Account created successfully.  
  
1. Create Account  
2. Login  
3. Deposit  
4. Withdraw  
5. Quit  
Enter your choice: 2  
Enter account number: 14785  
Enter your password: SARATH07  
Logged in as SARATH.  
  
1. Create Account
```

```
MiniProject Banking system  Version control  Current File  Run  Banking System.py  Banking System.py  Run  Banking System.py  5. Quit  
Enter your choice: 3  
Enter account number: 14785  
Enter the amount to deposit: 500  
Deposited $500.00. New balance: $1500.00  
  
1. Create Account  
2. Login  
3. Deposit  
4. Withdraw  
5. Quit  
Enter your choice: 4  
Enter account number: 14785  
Enter your password: SARATH07  
Enter the amount to withdraw: 250  
Withdrew $250.00. New balance: $1250.00  
  
1. Create Account  
2. Login  
3. Deposit  
4. Withdraw  
5. Quit  
Enter your choice: 5  
  
Process finished with exit code 0
```