Not syncing | GitHub - Search | New repository | clude ai - Search | Claude | Home | OOPS Banking assign

localhost:8889/notebooks/OOPS%20Banking%20assign.ipynb?

jupyter OOPS Banking assign Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[ ]:
```
#OOPS Banking
```

[ ]:
```
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 s
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
```

[ ]:
```
class BankAccount:
    def __init__(self):
        # Initialize with some dummy accounts for demonstration
        self.accounts = {
            '1001': {'pin': '1234', 'balance': 1000, 'name': 'John Doe'},
            '1002': {'pin': '5678', 'balance': 2000, 'name': 'Jane Smith'}
        }
        self.current_user = None

    def login(self, account_number, pin):
        """Authenticate user with account number and PIN"""
        if account_number in self.accounts and self.accounts[account_number]['pin'] == pin:
            self.current_user = account_number
            return True
        return False

    def deposit(self, amount):
        """Deposit money into the current account"""
        if not self.current_user:
            return "Please login first"
        if amount <= 0:
            return "Invalid amount"
```

ENG IN 09:56 PM 23-10-2024

jupyter OOPS Banking assign Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

```
            return "Please login first"
        if amount <= 0:
            return "Invalid amount"

        self.accounts[self.current_user]['balance'] += amount
        return f"Successfully deposited ${amount}. New balance: ${self.get_balance()}"

    def withdraw(self, amount):
        """Withdraw money from the current account"""
        if not self.current_user:
            return "Please login first"
        if amount <= 0:
            return "Invalid amount"
        if amount > self.accounts[self.current_user]['balance']:
            return "Insufficient funds"

        self.accounts[self.current_user]['balance'] -= amount
        return f"Successfully withdrew ${amount}. New balance: ${self.get_balance()}"

    def get_balance(self):
        """Check current balance"""
        if not self.current_user:
            return "Please login first"
        return self.accounts[self.current_user]['balance']

    def logout(self):
        """Logout current user"""
        self.current_user = None
        return "Logged out successfully"


def main():
    bank = BankAccount()

    while True:
        print("\n--- Welcome to Python Banking System ---")
```

jupyter OOPS Banking assign Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code JupyterLab Python 3 (ipykernel)

```
while True:
    print("\n=== Welcome to Python Banking System ===")
    print("1. Login")
    print("2. Deposit")
    print("3. Withdraw")
    print("4. Check Balance")
    print("5. Logout")
    print("6. Exit")

    choice = input("\nEnter your choice (1-6): ")

    if choice == '1':
        if bank.current_user:
            print("Already logged in!")
            continue

        acc_no = input("Enter account number: ")
        pin = input("Enter PIN: ")

        if bank.login(acc_no, pin):
            print(f"Welcome {bank.accounts[acc_no]['name']}!")
        else:
            print("Invalid account number or PIN")

    elif choice == '2':
        try:
            amount = float(input("Enter amount to deposit: $"))
            print(bank.deposit(amount))
        except ValueError:
            print("Invalid amount")

    elif choice == '3':
        try:
            amount = float(input("Enter amount to withdraw: $"))
            print(bank.withdraw(amount))
```

jupyter OOPS Banking assign Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code JupyterLab Python 3 (ipykernel)

```
            try:
                amount = float(input("Enter amount to withdraw: $"))
                print(bank.withdraw(amount))
            except ValueError:
                print("Invalid amount")

        elif choice == '4':
            balance = bank.get_balance()
            if isinstance(balance, (int, float)):
                print(f"Current balance: ${balance}")
            else:
                print(balance)

        elif choice == '5':
            print(bank.logout())

        elif choice == '6':
            print("Thank you for using Python Banking System!")
            break

        else:
            print("Invalid choice! Please try again.")


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

jupyter OOPS Banking assign Last Checkpoint: 13 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code JupyterLab Python 3 (ipykernel)

```python
if __name__ == "__main__":
    main()
```

```
=== Welcome to Python Banking System ===
1. Login
2. Deposit
3. Withdraw
4. Check Balance
5. Logout
6. Exit

Enter your choice (1-6):  1001
Invalid choice! Please try again.

=== Welcome to Python Banking System ===
1. Login
2. Deposit
3. Withdraw
4. Check Balance
5. Logout
6. Exit

Enter your choice (1-6):  1
Enter account number:  1001
Enter PIN:  1234
Welcome John Doe!

=== Welcome to Python Banking System ===
1. Login
2. Deposit
3. Withdraw
4. Check Balance
5. Logout
6. Exit

Enter your choice (1-6):  2
```

jupyter OOPS Banking assign Last Checkpoint: 13 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code JupyterLab Python 3 (ipykernel)

```
Enter your choice (1-6):  2
Enter amount to deposit: $ 1000
Successfully deposited $1000.0. New balance: $2000.0

=== Welcome to Python Banking System ===
1. Login
2. Deposit
3. Withdraw
4. Check Balance
5. Logout
6. Exit

Enter your choice (1-6):  3
Enter amount to withdraw: $ 500
Successfully withdrew $500.0. New balance: $1500.0

=== Welcome to Python Banking System ===
1. Login
2. Deposit
3. Withdraw
4. Check Balance
5. Logout
6. Exit

Enter your choice (1-6):  4
Current balance: $1500.0

=== Welcome to Python Banking System ===
1. Login
2. Deposit
3. Withdraw
4. Check Balance
5. Logout
6. Exit

Enter your choice (1-6):  5
```

jupyter OOPS Banking assign Last Checkpoint: 13 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code JupyterLab Python 3 (ipykernel)

```
6. Exit

Enter your choice (1-6):  4
Current balance: $1500.0

=== Welcome to Python Banking System ===
1. Login
2. Deposit
3. Withdraw
4. Check Balance
5. Logout
6. Exit

Enter your choice (1-6):  5
Logged out successfully

=== Welcome to Python Banking System ===
1. Login
2. Deposit
3. Withdraw
4. Check Balance
5. Logout
6. Exit

Enter your choice (1-6):  6
Thank you for using Python Banking System!
```

[ ]: