

Terminal App

T1A3



The Welcome function run when you first initialise the application. What this does is it checks to see if a csv file is created if not it will create the csv file and gives you a message db exist with a thumb up.

Create_Login function prompts you to input a user, pin, balance and it will save these into the csv file. While is used to start a loop Try and except used here to make sure you enter a integer into pin and a number into balance and break exits the loop.

Login function is using loops to check when you log in the user and pin you input matches a row in the csv file. If a match is found you will move on to the next step if not it prompts you to try again.

```
...
import csv
import emoji
from clear import clear
from colored import fg, bg, attr
from getpass4 import getpass
# name of csv
file_name = "login.csv"

# run to check if csv file exist if not then it create with preloaded header login, pin, balance and prints out db created
# if csv file exist it prints out db exist
def welcome():
    try:
        db = open(file_name, "r")
        db.close()
        print(emoji.emojize('db exist:thumbs_up:'))
    except FileNotFoundError as e:
        db = open(file_name, "w")
        db.write("login, PIN, Balance\n")
        db.close()
        print(emoji.emojize('db created:thumbs_up:'))

# input user, pin, balance gets written to csv file
# accepts str, int and float for inputs
# checks if input is valid without crashing the app using try and except
def create_login(file_name):
    while True:
        try:
            user = input("Enter new Username: ").lower()
            pin = int(getpass("Enter new PIN: ").strip())
            balance = float(input("Enter new balance: "))
            with open(file_name, "a", newline="") as f:
                writer = csv.writer(f, delimiter=",")
                writer.writerow([user, pin, balance])
                break
        except:
            print(emoji.emojize('Something went wrong.:thumbs_down:'))

# checks if log in details entered here matches created_login
# for loops used to check through each row for a match of user and pin
def login(file_name):
    while True:
        try:
            user = input("Enter Username: ").lower()
            pin = int(getpass("Enter PIN: ").strip())
            with open(file_name, "r", newline="") as f:
                reader = csv.reader(f, delimiter=",")
                for row in reader:
                    if row[0] == user and int(row[1]) == pin:
                        print(emoji.emojize('Login successful:thumbs_up:'))
                        return True, float(row[2])
                print(emoji.emojize('Invalid username or pin. Please try again.:thumbs_down:'))
        except:
            print(emoji.emojize('Something went wrong.:thumbs_down:'))
```

Create_menu function prints out a menu and ask you to enter a option.

While loop is here to and will stop only when true in this case if 4 is selected this loop will close. If/elif is used to check the choice in between 1-3 and depending on the choices selected it will prompt the user to enter something.

```
# main menu
def create_menu():
    print(f'{fg(1)}\t*****{attr(0)}')
    print(f'{fg(1)}\t***  Welcome To CA ATM  ***{attr(0)}')
    print(f'{fg(1)}\t*****{attr(0)}')
    print(f'{fg(1)}{bg(7)}1. Check balance\n{attr(0)}')
    print(f'{fg(2)}{bg(7)}2. Withdraw\n{attr(0)}')
    print(f'{fg(14)}{bg(7)}3. Deposit\n{attr(0)}')
    print(f'{fg(166)}{bg(7)}4. Exit\n{attr(0)}')
    choice = input("Enter your selection: ")
    return choice
```

```
choice = ""

while choice != "4":
    choice = create_menu()

    if choice == "1":
        username = input("Enter your username: ")
        balance = get_balance(username)
        if balance is not None:
            print(f'Your balance is: {fg(0)}{bg(7)}{balance}{attr(0)}')
    elif choice == "2":
        amount = float(input("Enter withdrawal amount: "))
        withdraw(username, amount)
    elif choice == "3":
        amount = float(input("Enter Deposit amount: "))
        deposit(username, amount)
    elif choice == "4":
        print(emoji.emojize('Thank you, and have a nice day.:full_moon_face:'))
        break
    else:
        print(emoji.emojize('Invalid selection. Please try again.:frowning_face_with_open_mouth:'))
```

Get_balance function will read the balance for the current log in user

Update_balance function will update the current balance once a withdrawal/deposit is made.

Withdraw gets the balance from the current user and calculates the balance and updates it to the update balance function. If and else is used to prevent you from withdrawing more than your balance.

Deposit functions the same as withdraw but takes away money from your current balance.

```
# balance function for main menu check csv file and matches user and balance
def get_balance(username):
    with open(file_name, "r", newline="") as f:
        reader = csv.reader(f, delimiter=",")
        for row in reader:
            if row[0] == username:
                return float(row[2])
    print("Invalid user.")
    return None

# update balance function is used to when changes are made to get_balance by either withdraw or deposit function below
def update_balance(username, new_balance):
    rows = []
    with open(file_name, "r", newline="") as f:
        reader = csv.reader(f, delimiter=",")
        for row in reader:
            if row[0] == username:
                row[2] = str(new_balance)
            rows.append(row)

    with open(file_name, "w", newline="") as f:
        writer = csv.writer(f, delimiter=",")
        writer.writerows(rows)

# withdraw - current balance and display new balance
# doesn't allow you to withdraw funds that exceed your balance
def withdraw(username, amount):
    balance = get_balance(username)
    if balance is not None:
        print(f"You, 8 hours ago • added withdraw function ...")
        if balance >= amount:
            new_balance = balance - amount
            update_balance(username, new_balance)
            print(f'Withdrawal of {fg(0)}{bg(7)}{amount}{attr(0)} successful.')
        else:
            print(emoji.emojize('Insufficient funds.:frowning_face_with_open_mouth:'))
    else:
        print(emoji.emojize('Invalid user:frowning_face_with_open_mouth:'))

# Deposit funds
def deposit(username, amount):
    balance = get_balance(username)
    if balance is not None:
        new_balance = balance + amount
        update_balance(username, new_balance)
        print(f'Deposit of {fg(0)}{bg(7)}{amount}{attr(0)} successful.')
    else:
        print(emoji.emojize('Invalid user:frowning_face_with_open_mouth:'))
```

Challenges

Csv file - read, match

