Login Register Readme File

User Registration and Login System

TASKS

1.Allow users to register with a unique username and a password.
2. Store user credentials securely (initially, a simple dictionary can be used).
3. Prevent duplicate registrations for the same username.
4. Provide a login mechanism where users can log in using their registered username and password.
5. Validate login credentials and provide appropriate messages for success or failure.
6. Implement a loop to allow multiple registrations or logins until the user exits.

STEPS

1. Plan Data Storage

- Decide on a method to store user credentials (e.g., a dictionary where usernames are keys and passwords are values).

2. Create the Registration Feature

- Prompt the user to enter a username.
- Check if the username already exists in the data storage.
- If it exists, inform the user and ask for a different username.
- If it doesn't exist, proceed to the next step.
- Prompt the user to enter a password.
- Store the username and password in the data storage.
- Display a success message indicating the registration is complete.

3. Create the Login Feature

- Prompt the user to enter their username.
- Check if the username exists in the data storage.
- If it doesn't exist, inform the user that the username is not registered.
- If it exists, proceed to the next step.
- Prompt the user to enter their password.
- Validate the entered password against the stored password.
- If the password matches, display a success message.
- If the password doesn't match, inform the user and allow them to try again.

4. Design the User Menu

- Create a menu-driven interface with options for:
- Registering a new user.
- Logging in as an existing user.
- Exiting the program.

- Prompt the user to choose an option.
- Based on the user's choice, invoke the appropriate functionality (registration or login).
- Allow the program to run in a loop until the user explicitly chooses to exit.

5. Implement Input Validation

- Ensure that the username and password inputs are non-empty.
- Handle cases where the username contains invalid characters or is too long/short.
- Prevent users from entering spaces or special characters in the username if required.

6. Test the System

- Test registering multiple users to ensure no duplicate usernames are allowed.
- Test logging in with correct and incorrect credentials.
- Test edge cases, such as logging in with an unregistered username.
- Verify that the system exits gracefully when the exit option is selected.

```
d={}# Dictionary of users
#login Function
def Login():
    username=input("Enter Username ")
    if username in d:
        password=input("Enter Password ")
        if password==d[username]:
            print("--------------------Loggined Succefully \U0001F600")
        else:# password is not currect
            print('Password is incorrect Try Again')
            Login()
    else:#user is not available
        print("User doesn't Exist please Register")
        Register()

# Registration process
def Register():
    username=input("Enter Username: ")
    if (validate(username)):
        password=input("Enter Password:")
        d.update({username:password})
        print("-------------------------------Registration Succefull\U0001F600")

# Validating username
def validate(name):
    spchr=['!','@','#','$','%','^','&','*','(',')','-','~','+','//','\\']
    if len(name)==0:
        print('username should not be empty')
        Register()
    elif (len(name)>25):
        print('username not more than 25 chatercters')
        Register()
    elif any(c in spchr for c in name):
        print('username doesn\'t contain any special charecters')
        Register()
    elif name in d:
        print('Username is available enter other one')
        Register()
    else:
        return True
```

```
# here we are creating the menu for user with choice
# step 4th is implemented
def Menu(n):

   if n==1:
      Login()
   elif n==2:
      Register()
#

print()
print("**************Welcome To SBM Solutions ***************")
print()
print("Please Enter your Choice")
print("1.Login")
print("2.Register")
print("3.Exit")
n=int(input("Enter no: "))


while(n!=3):
   Menu(n)
   print("If you want to continue Please Enter your Choice if not press 3")
   if n!=3:
     print("1.Login")
     print("2.Resister")

   n=int(input("Enter no: "))
print("Thank you for Visiting \U0001F600")
```

Output:

Menu Display:

```
PS C:\Lavanya_Code\Python_Lectures_Assignments\Project_Login_Resister> python Login_Resister.py

**********Welcome To SBM Solutions **********

Please Enter your Choice
1.Login
2.Register
3.Exit
Enter no:
```

Existing Gracefully:

```
Enter no: 3
Thank you for Visiting 😀
```

Registration Test cases:

Test1: Empty Username given

```
Enter no: 2
Enter Username:
username should not be empty
Enter Username:
```

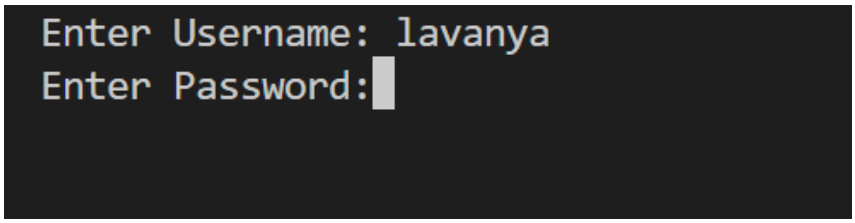
Test 2: Special Charecters in username

```
Enter Username: lavanya%la*
username doesn't contain any special charecters
Enter Username:
```

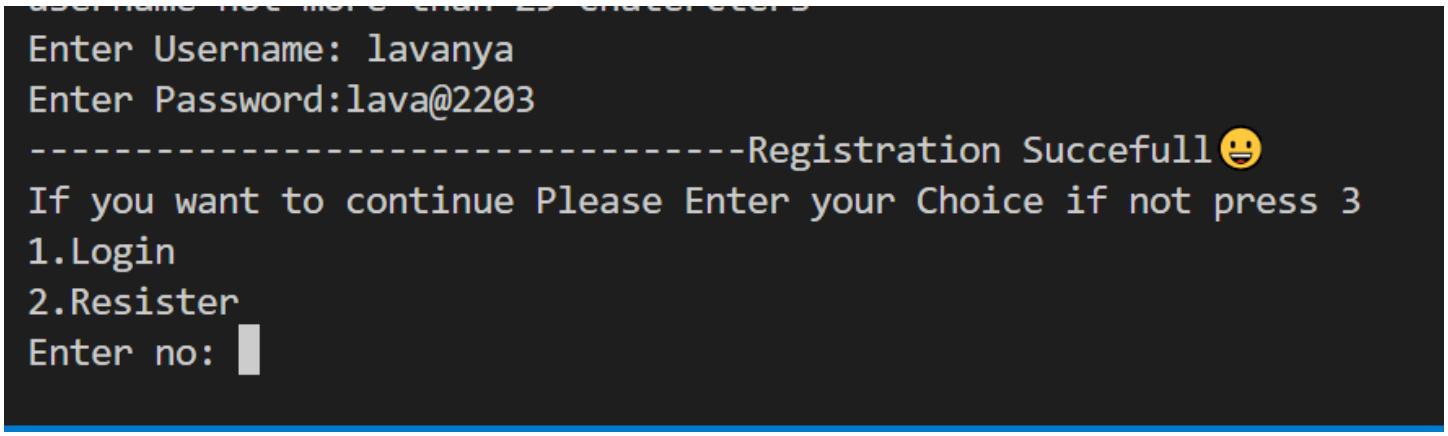
Test 3 : username is greater than 25 chars

```
Enter Username: 12345678901234567890123456
username not more than 25 chatercters
Enter Username:
```

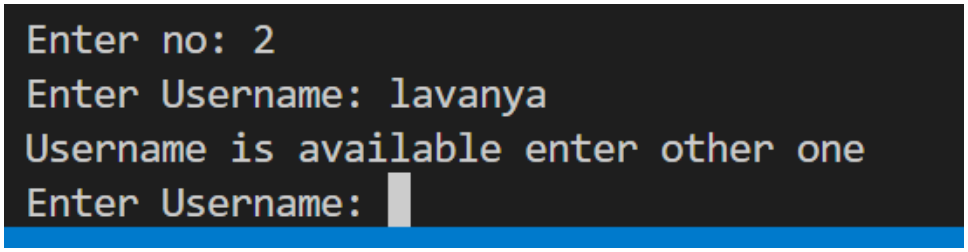
Test 4: username is currect



Test 5: Registration Successful



Test 6: Username already Exist



Login Test cases:

Test 1: username is not exist

```
Please Enter your Choice
1.Login
2.Register
3.Exit
Enter no: 1
Enter Username lavanya
User doesn't Exist please Register
Enter Username: lavanya
Enter Password:lava@2203
---------------------------------Registration Succefull😀
If you want to continue Please Enter your Choice if not press 3
1.Login
2.Resister
Enter no: 1
Enter Username lavanya
Enter Password lava@2203
--------------------Loggined Succefully 😀
If you want to continue Please Enter your Choice if not press 3
```

Test 2: password Doesn't match

```
1.Login
2.Resister
Enter no: 1
Enter Username lavanya
Enter Password lava@codnera
Password is incorrect Try Again
Enter Username
```

Test 3: Loggined succefully

```
Enter Username lavanya
Enter Password lava@2203
--------------------Loggined Succefully 😀
If you want to continue Please Enter your Choice if not press 3
```

To Do List Application

```
print("***************************** To Do List
************************************")

d={}
def Add():
   length=int(input("Enter no.of Items you want to add "))
   for i in range(length):
      item=input("Enter Item ")
      d.update({len(d)+1:item})
   print("Items Added Succefully .....")

# deleting Items
def Delete():
   # When List is Empty
   if len(d)==0:
      print("List is empty")
      return

   dlen=int(input('Enter no.of Items you want to delete '))
   # when no of items are not present in list
   if dlen>len(d):
      print("No of Items Not Present ")
      return

   else:
    while(dlen!=0):
      dno=int(input("Enter Task Number "))
      if dno not in d:
         print('Task Not exist')

      else:
         d.pop(dno)
         dlen-=1
   print("-------------------------Deleted Succefully \U0001F600")



# Display Items
```

```
def Display():
    print("----------------------Your To Do List-------------------")
    # when list is empty
    if len(d)==0:
        print("***************No Tasks To do \U0001F600")
    for k,v in d.items():
        print(str(k),"-",v,"")


# Menu Programe
def Menu(n):
    if n==1:
        Add()
    elif n==2:
        Delete()
    elif n==3:
        Display()


print("Please Enter your Choice: ")
print('1.Add Items')
print('2.Delete Items')
print('3.Display List')
print('4.Exist')
n=int(input('Enter Choice: '))
while(n!=4):
    Menu(n)
    print('if you want to continue enter choice if not press 4')
    if (n!=4):
        print('1.Add Items')
        print('2.Delete Items')
        print('3.Display List')
    n=int(input("Enter Choice: "))
print("Thank you for Visiting \U0001F600")
```

To Do List Application

Output

Test 1: Menu

```
PS C:\Lavanya_Code\Python_Lectures_Assignments\To_Do_List_Application> python To_Do_List_App.py
************************** To Do List ***********************************
Please Enter your Choice:
1.Add Items
2.Delete Items
3.Display List
4.Exist
Enter Choice:
```

Test 2: Exist From Menu

```
Please Enter your Choice:
1.Add Items
2.Delete Items
3.Display List
4.Exist
Enter Choice: 4
Thank you for Visiting 😀
PS C:\Lavanya_Code\Python_Lectures_Assignments\To_Do_List_Application>
```

Add Items

Test 1: Add Items

```
1.Add Items
2.Delete Items
3.Display List
4.Exist
Enter Choice: 1
Enter no.of Items you want to add 3
Enter Item yoga
Enter Item meditation
Enter Item cooking
Items Added Succefully .........
if you want to continue enter choice if not press 4
```

Display Items

Test 1: List is empty

```
Please Enter your Choice:
1.Add Items
2.Delete Items
3.Display List
4.Exist
Enter Choice: 3
----------------------Your To Do List-------------------
**************No Tasks To do 😀
```

Test 2: List having Items

```
Enter no.of Items you want to add 3
Enter Item wake up at 6
Enter Item meditation
Enter Item yoga
Items Added Succefully ..........
if you want to continue enter choice if not press 4
1.Add Items
2.Delete Items
3.Display List
Enter Choice: 3
----------------------Your To Do List-------------------
1 - wake up at 6 ✅
2 - meditation ✅
3 - yoga ✅
if you want to continue enter choice if not press 4
```

Delete Items

Test 1: When no Items in list

```
1.Add Items
2.Delete Items
3.Display List
Enter Choice: 2
List is empty
```

Test 2: When no.of Items are not present in list

```
----------------------Your To Do List-------------------
1 - wake up at 6 ✅
2 - meditation ✅
3 - yoga ✅
if you want to continue enter choice if not press 4
1.Add Items
2.Delete Items
3.Display List
Enter Choice: 2
Enter no.of Items you want to delete 6
No of Items Not Present
```

Test 3: When Tasks are not exist

```
1.Add Items
2.Delete Items
3.Display List
Enter Choice: 2
Enter no.of Items you want to delete 2
Enter Task Number 13
Task Not exist
```

Test 4: Deleted succefully

```
Enter Choice: 2
Enter no.of Items you want to delete 2
Enter Task Number 13
Task Not exist
Enter Task Number 1
Enter Task Number 2
--------------------------Deleted Succefully 😀
```

Bank Account Management System

3. Bank Account Management System

Tasks:

1. Start with an initial balance of 0.
2. Implement a deposit function to add a specific amount to the account.
3. Implement a withdraw function to deduct a specific amount from the account, ensuring sufficient balance.
4. Provide a function to check the current account balance.
5. Add error handling for invalid amounts (e.g., negative or non-numeric input).
6. Create a menu-driven interface to switch between deposit, withdraw, check balance, and exit.

Code

```
amount=0 #Start with an initial balance of `0`

#Implement a deposit function to add a specific amount to the account
def Deposit():
   global amount
   ea=int(input('Enter Amount '))
   if ea<0:
      print("please Enter Valid Number")
      Deposit()
   else:
    amount+=ea

# Implement a withdraw function to deduct a specific amount from the account, ensuring sufficient balance
def Withdraw():
   global amount
   wa=int(input("please enter amount "))
   if wa>amount:
      print("Insuffisient Balance")
```

```
        Withdraw()
    else:
        amount-=wa


# Provide a function to check the current account balance
def CheckBalance():
    global amount
    print("Your Current Balance:",amount)

n=0
print("Please choose you Choice")
print("1.Deposit")
print("2.Withdraw")
print("3.CheckBalance")
print("4.Exist")
n=int(input("Enter Your Chioce "))

# Create a menu-driven interface to switch between deposit, withdraw, check balance, and
exit.
def Menu(n):
    if n==1:
        Deposit()
    elif n==2:
        Withdraw()
    elif n==3:
        CheckBalance()



while(n!=4):
    Menu(n)
    print("If you want to continue press y if not press n")
    c=input()
    if c=='y':
      print("1.Deposit")
      print("2.Withdraw")
      print("3.CheckBalance")
      n=int(input("Enter you choice"))

    else:
        n=4
print("Thank you for Visiting \U0001F600")
```

Output

Test 1: Menu Preview

```
Please choose you Choice
1.Deposit
2.Withdraw
3.CheckBalance
4.Exist
Enter Your Chioce 4
Thank you for Visiting 😀
PS C:\Lavanya_Code\Python_Lectures_Assignments\Bank_Account_Management_System>
```

Test 2: Deposit entering negative amount

```
Please choose you Choice
1.Deposit
2.Withdraw
3.CheckBalance
4.Exist
Enter Your Chioce 1
Enter Amount -100
please Enter Valid Number
Enter Amount
```

Test 3: Deposit Entering Valid amount

```
please Enter Valid Number
Enter Amount 1000
If you want to continue press y if not press n
```

Test 4 :Check Balance

```
1.Deposit
2.Withdraw
3.CheckBalance
Enter you choice3
Your Current Balance: 1000
```

Test 5: Withdraw insufficient balance

```
1.Deposit
2.Withdraw
3.CheckBalance
Enter you choice2
please enter amount 2000
Insuffisient Balance
please enter amount
```

Test 6: Entering Sufficient Balance

```
Enter you choice2
please enter amount 2000
Insuffisient Balance
please enter amount 500
If you want to continue press y if not press n
```

Check balance after withdraw done

```
y
1.Deposit
2.Withdraw
3.CheckBalance
Enter you choice3
Your Current Balance: 500
```

4. Library Management System

Tasks:
- 1. Maintain a list of available books along with their copies (e.g., a dictionary with book names as keys and copies as values).
- 2. Provide an option to add new books or update the number of copies for existing books.
- 3. Display a list of all available books with the number of copies for each.
- 4. Allow users to borrow books, reducing the available copies by 1.
- 5. Prevent users from borrowing books that are out of stock.
- 6. Include input validation and a loop to allow repeated operations until the user exits.

Code

```
# Maintain a list of available books along with their copies (e.g., a dictionary with book names as keys and copies as values).

Book_List={
   "The Power of Now": 10,
   "Atomic Habits": 15,
   "The Four Agreements": 9,
   "The Alchemist": 18,
   "You Are a Badass": 10,
   "The Power of Habit": 12,
   "Start with Why": 11,
   "The Lean Startup": 7,
   "The 48 Laws of Power": 6,
   "Deep Work": 8,
   "Daring Greatly": 11,
   "The Secret": 13,
   "The Magic of Thinking Big": 8,
   "The Slight Edge": 7,
   "The 5 AM Club": 10,
   "Becoming": 9,
   "The Science of Getting Rich": 5,
   "Principles: Life and Work": 6,
   "The Compound Effect": 7,
   "The Success Principles": 8,
   "Awaken the Giant Within": 12,
   "You Can Heal Your Life": 10,
   "The Untethered Soul": 13,
   "The One Thing": 11,
   "The Confidence Gap": 8,
```

```
    "Who Moved My Cheese?": 6,
    "The Motivation Manifesto": 5,
    "The Big Leap": 7,
    "The Little Book of Hygge": 7,
    "Rising Strong": 9,
    "The Confidence Code": 8,
    "The Motivation Hacker": 7
}


# Provide an option to add new books or update the number of copies for existing books.
def Add_Books():
    global Book_List
    l=int(input("Enter No.of books you want to add "))
    for i in range(l):
        name=input("Enter Name Of book ")
        QTY=int(input("enter Quantity of book "))
        if name in Book_List:
            QTY=Book_List[name]+QTY
        else:
            QTY=QTY
        Book_List.update({name:QTY})



# Display a list of all available books with the number of copies for each.
def Display_Available_Books():
    global Book_List
    for k,v in Book_List.items():
        print(k,v)


# Allow users to borrow books, reducing the available copies by 1.
def Borrow_Books():
    ans=int(input("Enter No. of book you want to borrow "))
    for i in range(ans):
        name=input("Enter name of book ")
        QTY=int(input("Enter Quntity "))
        # Prevent users from borrowing books that are out of stock.
        if name not in Book_List.keys():
            print("Entered book is out of Stock")
            continue
```

```
        else:
            Book_List[name]=Book_List[name]-QTY

# Menu
def Menu(n):

    if n==1:
        Add_Books()
    elif n==2:
        Display_Available_Books()
    elif n==3:
        Borrow_Books()
print("Select Your Choice")
print("1.Add_Books")
print("2.Display_Available_Books")
print("3.Borrow _Books")
print("4.Exist")

n=int(input("Enter Your Choice "))
while(n!=4):
    Menu(n)
    ans=input("if you want to continue press y if not press n ")
    if ans=='y':
        print("1.Add_Books")
        print("2.Display_Available_Books")
        print("3.Borrow _Books")
        n=int(input("Enter Your Choice "))
    else:
        n=4
print("Thank you For Visiting \U0001F600")
```

Output

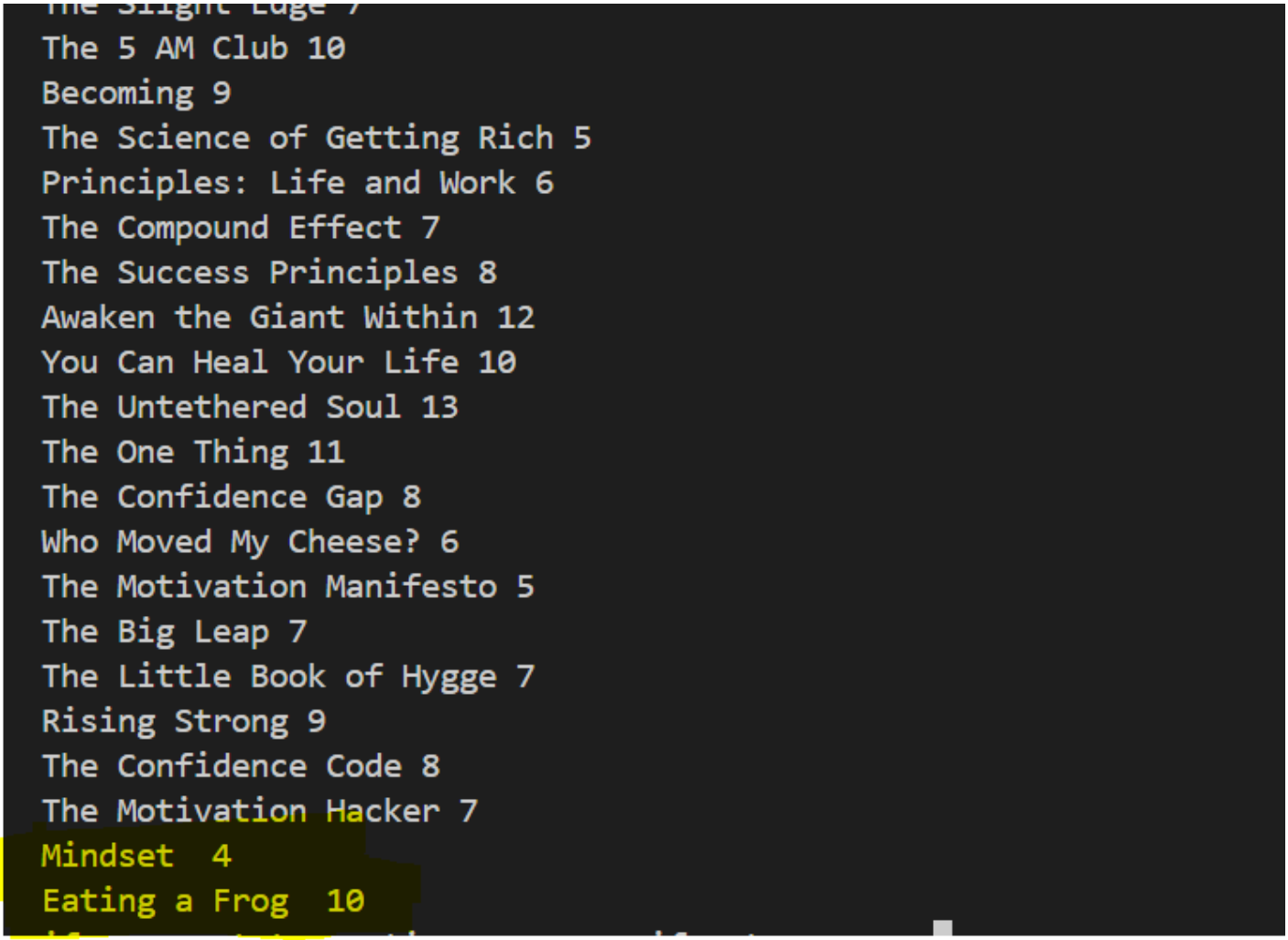
Test 1: Menu Test

```
PS C:\Lavanya_Code\Python_Lectures_Assignments\Library_Management_
Select Your Choice
1.Add_Books
2.Display_Available_Books
3.Borrow _Books
4.Exist
Enter Your Choice 4
Thank you For Visiting 😀
```

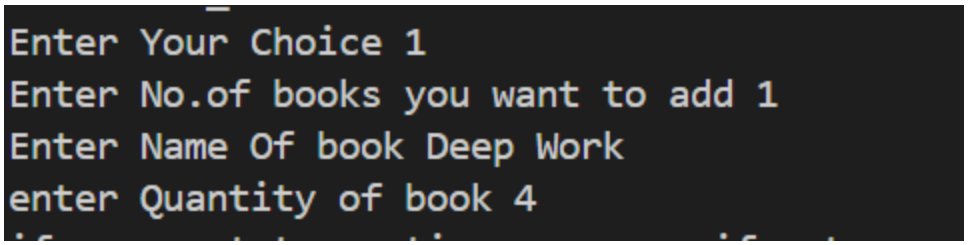
Test 2 :Adding New books

```
Enter Your Choice 1
Enter No.of books you want to add 2
Enter Name Of book Mindset
enter Quantity of book 4
Enter Name Of book Eating a Frog
enter Quantity of book 10
if you want to continue press y if not press n
```

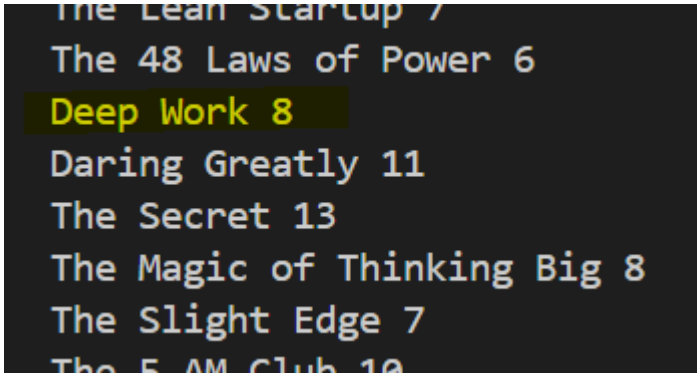
Test 3: Displaying Available all books



Test 4: adding available books



Before adding book QTY



After adding book QTY

```
The 48 Laws of Power 6
Deep Work 12
Daring Greatly 11
The Secret 13
The Magic of Thinking Big
```

Test 6: Borrow Books Out of stock

```
Enter Your Choice 3
Enter No. of book you want to borrow 2
Enter name of book lavanya
Enter Quntity 3
Entered book is out of Stock
Enter name of book
```

Test 7: Enter valid book name

```
Enter name of book Deep Work
Enter Quntity 4
if you want to continue press y if not press n
```

Before borrow

```
The Lean Startup 7
The 48 Laws of Power 6
Deep Work 8
Daring Greatly 11
The Secret 13
The Magic of Thinking Big 8
The Slight Edge 7
The 5 AM Club 10
```

```
The 48 Laws of Power 6
Deep Work 4
Daring Greatly 11
The Secret 13
The Magic of Thinking Big 8
```

Quiz-Application

```
# Prepare a list of questions and their correct answers.
d={1:{'What is a correct syntax to output "Hello" in Python?':
     ['Print("Hello")','print(Hello)','print Hello','print("Hello")'],
     'ans':4
     },

   2:{'Is Python code compiled or interpreted?':['Python code is both compiled and
interpreted',' Python code is neither compiled nor interpreted','Python code is only
compiled','Python code is only interpreted'],
       'ans':1
       },
   3:{'All keywords in Python are in _____':['Capitalized','lower case','UPPER CASE','None
of the mentioned'],
         'ans':4},
   4:{'What is the order of precedence in python?':[' Exponential, Parentheses, Multiplication,
Division, Addition, Subtraction','Parentheses, Exponential, Multiplication, Division, Addition,
Subtraction','Exponential, Parentheses, Division, Multiplication, Addition,
Subtraction','Parentheses, Exponential, Multiplication, Addition, Division, Subtraction'],
        'ans':2},

   5:{'What does pip stand for python?':['Pip Installs Python','Pip Installs Packages','Preferred
Installer Program','All of the mentioned'],
      'ans':3},
   6:{' Which of the following is the truncation division operator in Python?':['/','//','%','|'],
      'ans':2},
   7:{'What will be the output of the following Python function?\nmin(max(False,-3,-4),
2,7)':[2,-4,0,7],
      'ans':3},
   8:{'What arithmetic operators cannot be used with strings in Python?':['+','*','-','All Of
Above'],'ans':3
      },
   9:{'Which of the following statements is used to create an empty set in
Python?':['()','[]','{}','set()'],
      'ans':4},
   10:{'What is the maximum possible length of an identifier in Python?':['71','32','64','None Of
All'],'ans':4}


        }
```

```
# Create a function to display each question and accept the user's answer.
def Quiz():
   n=1
   global Final_Score
   Final_Score=0
   score=0
   for val in d.values():

      print()
      for k,v in list(val.items()):
       if k!='ans':
         print(n,"-",k)
         print('1)',v[0])
         print('2)',v[1])
         print('3)',v[2])
         print('4)',v[3])
      n+=1
      an=int(input("Enter Your Answer "))
      if k=='ans':
       if an==v:
         score+=10 #Validate the user's answer and keep track of the score.
   Final_Score=((score/100)*100)
   return Final_Score



def View_Score(f_score):
   print("Your Recent Score is",f_score) # Allow users to view their most recent score without
retaking the quiz
   print("if you want to retake Quize please enter y",end=" ")#Provide an option to retake the
quiz and reset the score for a new attempt.
   out=input()
   if out=='y':
     f=Quiz()
     print("Your current Score: ",f) # after recaking providing current score
   else:
     return
```

```
Final_Score=0
def Menu(n):
    global Final_Score #setting as Global for keep tract of score
    if n==1:
        Final_Score=Quiz()
        print('Your Score:',Final_Score)# Display the final score after the quiz is completed
    elif n==2:
        View_Score(Final_Score)




print("******************Welcome To Quiz Computition*****************")
print('Please Enter your Choice')
print('1.Take Quiz')
print('2.View Score')
print('3.Exit')
n=int(input("Enter Choice "))
while(n!=3):
    Menu(n)
    print("If you want to continue press choice if not the press 3")
    if n!=3:
        print('1.Take quiz')
        print('2.View Score')
    n=int(input("Enter Choice "))

print("Thank you for visiting \u0001F600")
```

Output

Test 1: Menu Display

```
**********Welcome To Quiz Computition**********
Please Enter your Choice
1.Take Quiz
2.View Score
3.Exit
Enter Choice
```

Test 2: Final Score After Quiz is Completed

```
Enter Choice 1

1 - What is a correct syntax to output "Hello" in Python?
1) Print("Hello")
2) print(Hello)
3) print Hello
4) print("Hello")
Enter Your Answer 4

2 - Is Python code compiled or interpreted?
1) Python code is both compiled and interpreted
2)  Python code is neither compiled nor interpreted
3) Python code is only compiled
4) Python code is only interpreted
Enter Your Answer 1

3 - All keywords in Python are in _________
1) Capitalized
2) lower case
3) UPPER CASE
4) None of the mentioned
Enter Your Answer 4
```

```
4 - What is the order of precedence in python?
1)  Exponential, Parentheses, Multiplication, Division, Addition, Subtraction
2) Parentheses, Exponential, Multiplication, Division, Addition, Subtraction
3) Exponential, Parentheses, Division, Multiplication, Addition, Subtraction
4) Parentheses, Exponential, Multiplication, Addition, Division, Subtraction
Enter Your Answer 2

5 - What does pip stand for python?
1) Pip Installs Python
2) Pip Installs Packages
3) Preferred Installer Program
4) All of the mentioned
Enter Your Answer 3

6 -  Which of the following is the truncation division operator in Python?
1) /
2) //
3) %
4) |
Enter Your Answer 2

7 - What will be the output of the following Python function?
min(max(False,-3,-4), 2,7)
1) 2
2) -4
3) 0
4) 7
Enter Your Answer 3
```

```
8 - What arithmetic operators cannot be used with strings in Python?
1) +
2) *
3) -
4) All Of Above
Enter Your Answer 4

9 - Which of the following statements is used to create an empty set in Python?
1) ()
2) []
3) {}
4) set()
Enter Your Answer 4

10 - What is the maximum possible length of an identifier in Python?
1) 71
2) 32
3) 64
4) None Of All
Enter Your Answer 4
Your Score: 90.0
```

Test 3: View Score Recent Score and without Retaking

```
If you want to continue press choice if not the press 3
1.Take quiz
2.View Score
Enter Choice 2
Your Recent Score is 90.0
if you want to retake Quize please enter y
```

Test 3: providing Option For Rekaing Quiz

```
Your Recent Score is 90.0
if you want to retake Quize please enter y y

1 - What is a correct syntax to output "Hello" in Python?
1) Print("Hello")
2) print(Hello)
3) print Hello
4) print("Hello")
Enter Your Answer 4

2 - Is Python code compiled or interpreted?
1) Python code is both compiled and interpreted
2)  Python code is neither compiled nor interpreted
3) Python code is only compiled
4) Python code is only interpreted
Enter Your Answer 1

3 - All keywords in Python are in _________
1) Capitalized
2) lower case
3) UPPER CASE
4) None of the mentioned
Enter Your Answer 4
```

```
4 - What is the order of precedence in python?
1)  Exponential, Parentheses, Multiplication, Division, Addition, Subtraction
2) Parentheses, Exponential, Multiplication, Division, Addition, Subtraction
3) Exponential, Parentheses, Division, Multiplication, Addition, Subtraction
4) Parentheses, Exponential, Multiplication, Addition, Division, Subtraction
Enter Your Answer 2

5 - What does pip stand for python?
1) Pip Installs Python
2) Pip Installs Packages
3) Preferred Installer Program
4) All of the mentioned
Enter Your Answer 3

6 -  Which of the following is the truncation division operator in Python?
1) /
2) //
3) %
4) |
Enter Your Answer 2

7 - What will be the output of the following Python function?
min(max(False,-3,-4), 2,7)
1) 2
2) -4
3) 0
4) 7
Enter Your Answer 3
```

```
8 - What arithmetic operators cannot be used with strings in Python?
1) +
2) *
3) -
4) All Of Above
Enter Your Answer 3

9 - Which of the following statements is used to create an empty set in Python?
1) ()
2) []
3) {}
4) set()
Enter Your Answer 4

10 - What is the maximum possible length of an identifier in Python?
1) 71
2) 32
3) 64
4) None Of All
Enter Your Answer 4
Your current Score:  100.0
```

After retaking current Score is 100.

Test 4: after retaking your currect score is provided as recent score and giving option to retake quiz?

```
Your current Score:  100.0
If you want to continue press choice if not the press 3
1.Take quiz
2.View Score
Enter Choice 2
Your Recent Score is 100.0
if you want to retake Quize please enter y
```

Test 5:Exist From Code

```
Enter Choice 3
Thank you for visiting 😀
PS C:\Lavanya_Code\Python_Lectures_Assignments\Quiz_app>
```