**Book Inventory Management System**

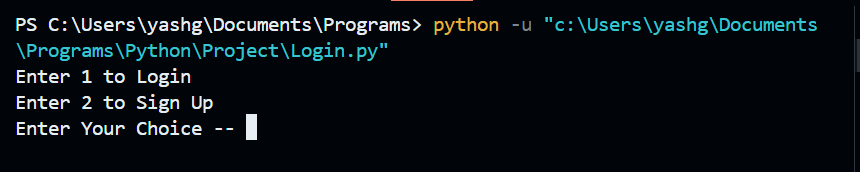
**This program deal with two types of user:**

1. **Admin**
2. **Customer**

**In order to archive maximum readability the whole program is divided into several file along with two separate txt file which store data . All the files are listed down along with their functions:-**

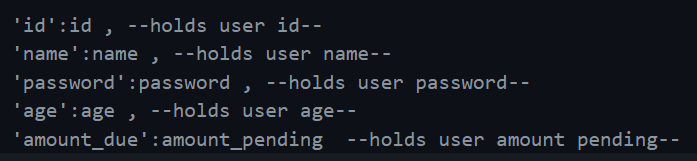
1. **login.py---** This is the python source code for providing a login interface . This is also the starting point of the program .
2. **user\_data\_handler.py---** This python source code provides all the functions to handle user data and handles login interface.
3. **user\_handler.py**---This python sources provides an interface to a Customer and handles all the customer related requests.
4. **books\_data\_handler.**py—This python source code provides all the functions to handle book data. It also provides interface to admin to deal with books information and an interface for customer to buy books .
5. **admin\_handler.**py—This python source code provides an interface to admin and deals with all his requests
6. **user.**txt—This text file store data of users.
7. **books.**txt—This txt file store data of books
8. **analysis.py – this python source manages analysis of book data**

**Login.py**

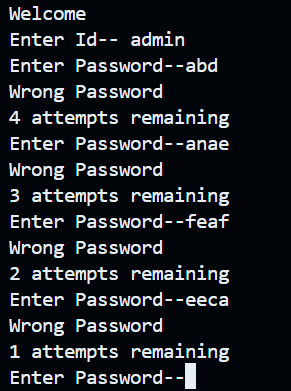
****

**user\_data\_handler.py**

1. **user\_database=[ ]**--It holds a list of dictionary data type designed to store user data.



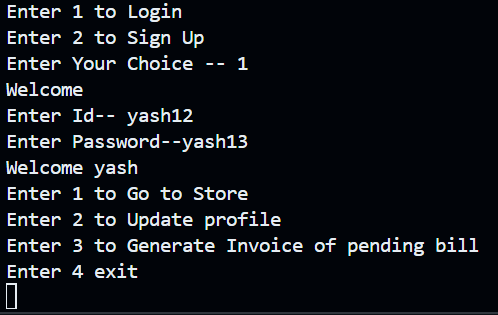
1. def login(): It provide an interface for login irrespective of user type. It cross-checks user input information with the user database and returns id if the login is successful .It deals with brute force attack by limiting a number of attempts.



1. def check\_id(s)->int: -- This functions check if the id is present in user database or not . if present then returns the index of the user in userdatabase
2. def check\_password(password,index)->bool: this funtions checks if password is correct or not for a user
3. def get\_data( ): This reads data from file “user.txt”
4. def fill\_user\_database(line): This function receives line from get\_data function and fill data in “user\_database”
5. def add\_user(): This functions add a new user in “user\_database”.
6. def remove\_user(): It removes a user from "user\_database”
7. def update\_user\_information(id): It updates user information of a specific user in "user\_database"
8. def set\_bill\_for\_user(userid,amount): Set pending amount for a user .
9. def get\_user\_pending\_bill(userid)->int : returns the pending amount of a user.

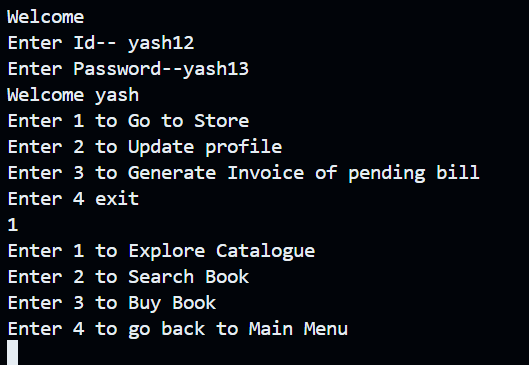
**user\_handler.py**

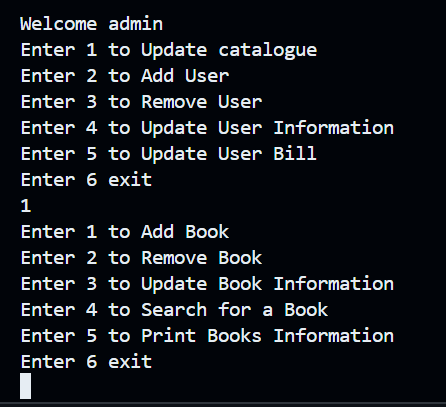
This code provides an interface to user to deal with all requests.



**books\_data\_handler.py**

It contains all the functions related to handle books data .It also provides interface for admin and customer.

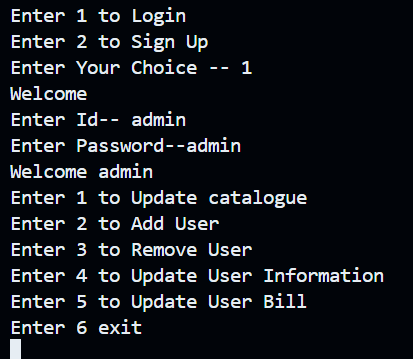




1. books=[] This list store books information in a dictionary.
2. def get\_data(): This function read data from the file “books.txt”
3. def get\_book\_data(line): This function receive line from “get\_data()” function and fill the list “books” by filtering data from the received line.
4. def add\_book():This function adds a new book dictionary in list "books"
5. def print\_book\_details():This function prints information of all the books present in list "books"
6. def remove\_book():This function remove a books form list "books"
7. def update\_book\_details(): This function updates books information.
8. def search\_book():This functions search if a book is present in the list "books"
9. def fill\_book\_data():This function writes the books information in file "books.txt"
10. def books\_data\_controller\_for\_admin():this function provides an interface for admin to deal with all his requests
11. def books\_data\_controller\_for\_user(id): This function provides an interface for customer and deals with all his requests
12. def buybook(user\_id):This provides an interface to customer to buy a book or to add book in a cart.
13. def check\_book\_id(book\_id:int)->bool: This function checks if the book id is valid or not
14. def checkout(user\_id:str , book\_cart):This function updates the pending bill of user as per the books he bought
15. def update\_inventory(book\_cart):This updates the “books”data when a user buys a book.

**admin\_handler.py**

This source code provides an interface for admin and deals with all his requests



**books.txt**

It holds data of all the books present in the inventory in a specific format.

“Book name , number of books available , price of book \n”

This format is used to store the data of books.

“ , ” in the format helps in filtering different types of information.

“\n” is added intentionally to make sure the data of different books don’t get mixed up.

**user.txt**

It holds data of all the users in a specific format.

“username , full name , password , age , amount due \n”

This format is used to store the data of users.

“ , ” in the format helps in filtering different types of information.

“\n” is added intentionally to make sure the data of different user don’t get mixed up.

**Analysis.py**

1. **Class analysis**
2. **Def most\_sold\_book() –** finds the most sold book
3. **Def trending\_books()—**finds top 5 sold book
4. **Def total\_sales() –** calculates total sales
5. **Def controller\_for\_user()—**manages options related to user
6. **Def controller –** manages options for admin