Library Management Model

Description

This project is a user-friendly GUI-based Model for managing a library.

Installation

Make sure that the `Github CLI` is properly installed on your device. After that, you're good to go ahead and clone the code into your system.

```
gh repo clone frankmathewsajan/library-management-system
```

Before executing the main script, ensure you have installed the necessary Python libraries. To do this, run the following commands in your terminal, making sure you're in the `root` of the project folder:

```
pip install -r requirements.txt
```

Usage

To run the script, navigate to the directory containing the script and run the following command:

```
python project.py
```

This will open a GUI authentication window asking for `Login` or `Register`. Follow the instructions in the window as it is.

Use the following credentials to log in for review purposes.

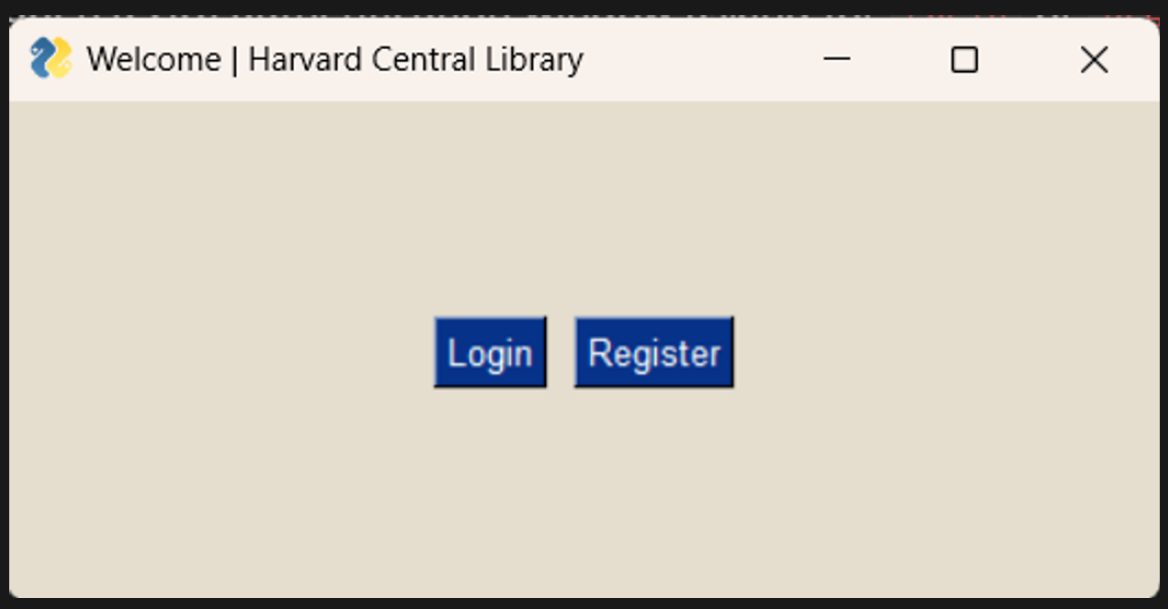
```
Username: FrankMathewSajan Password: Frank@2005
```

Tech Stack

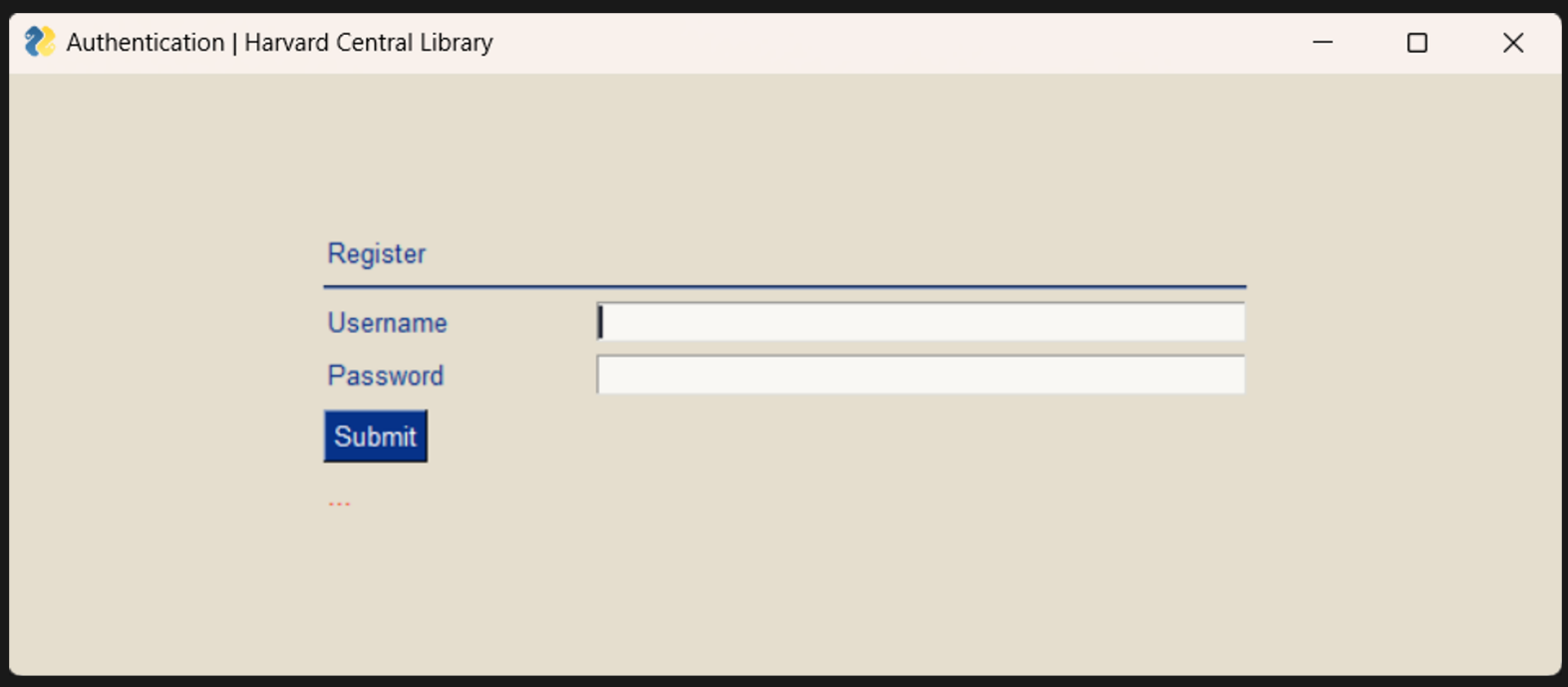
Front-End: `PySimpleGUI` , `Python`

Database: `SQLite3`

Screenshots and Output

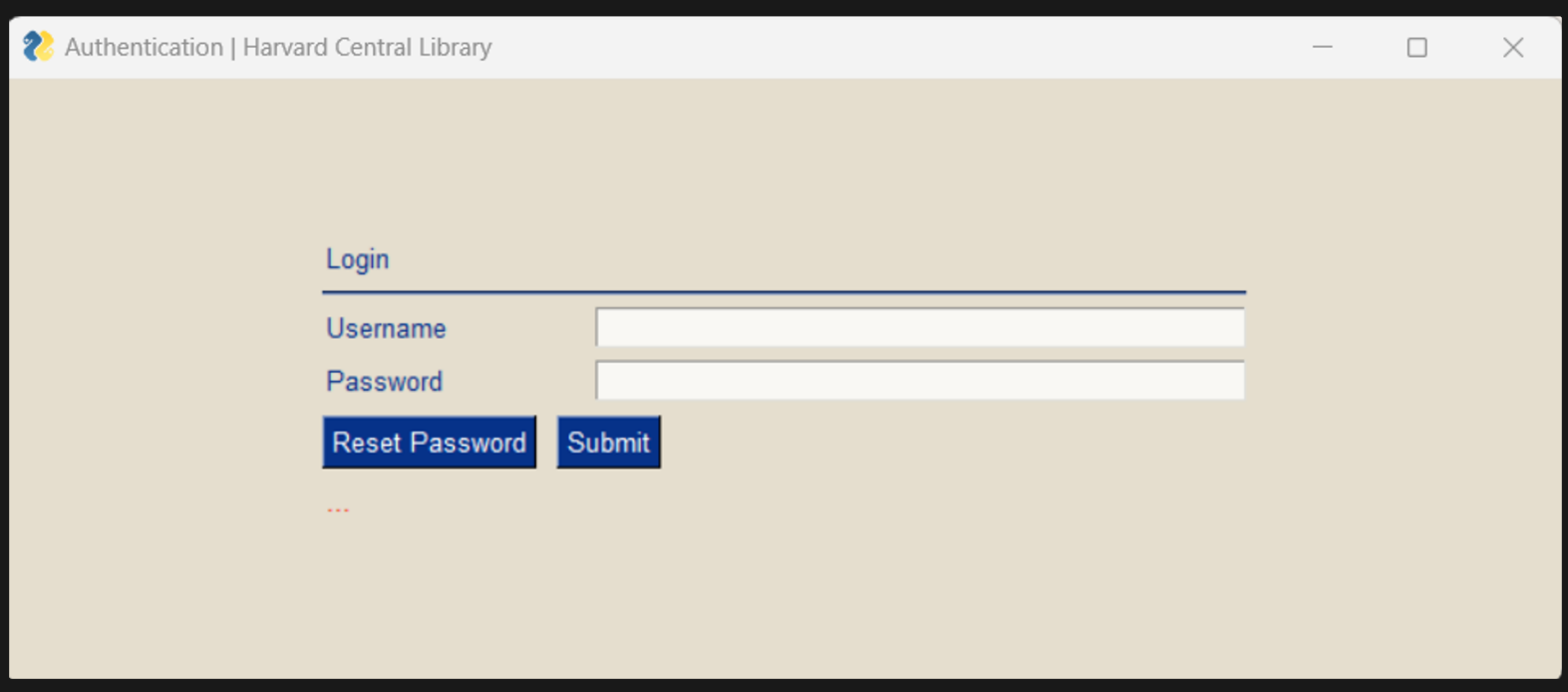


Welcome Page: If you're not logged in, you have options to either `Login` or `Register`.



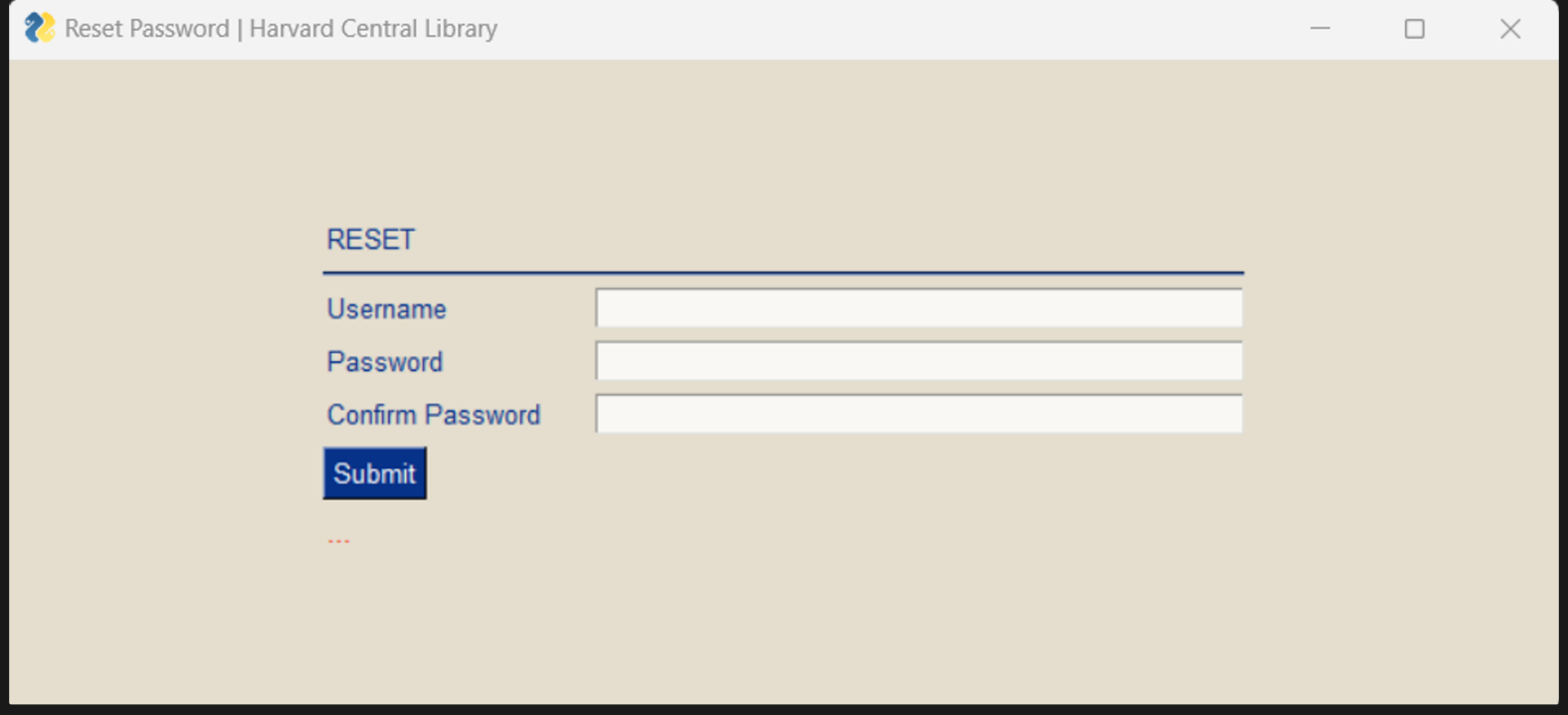
Shows this when the user is already logged in.

Register Page: You can register as a new user here.

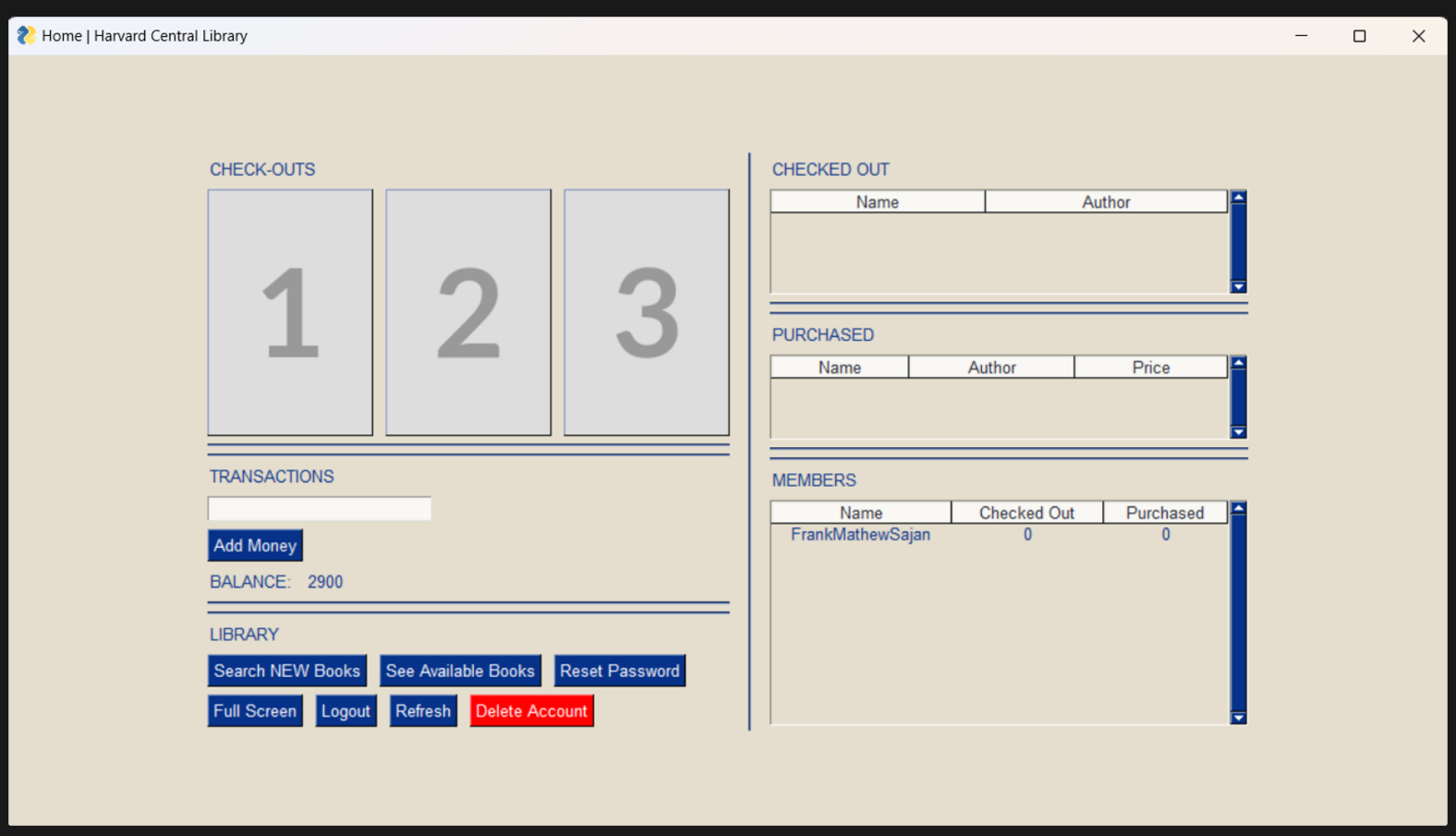


Login Page: Log in using the demo credentials or the newly registered credentials.

Reset Password Button: Opens a GUI for resetting the user's password.

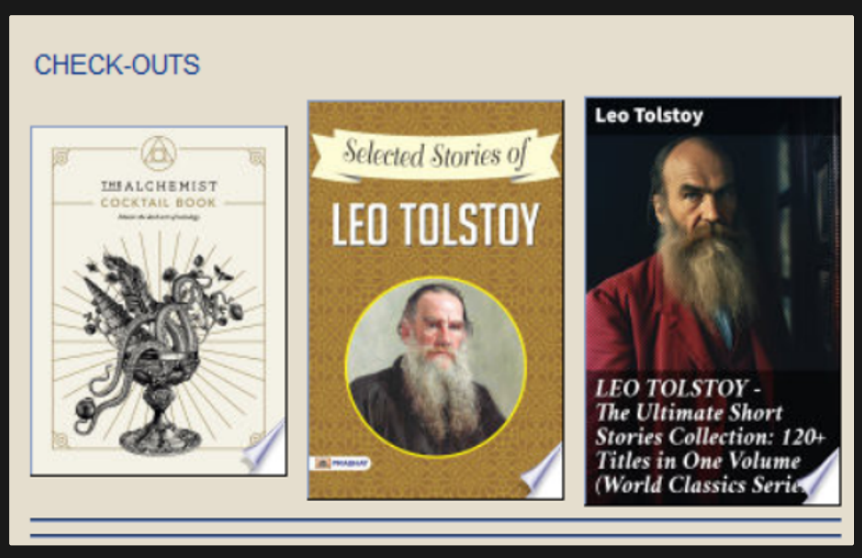


Reset Password Page: Let the user change their password, assuming the account exists, and the new password is confirmed.

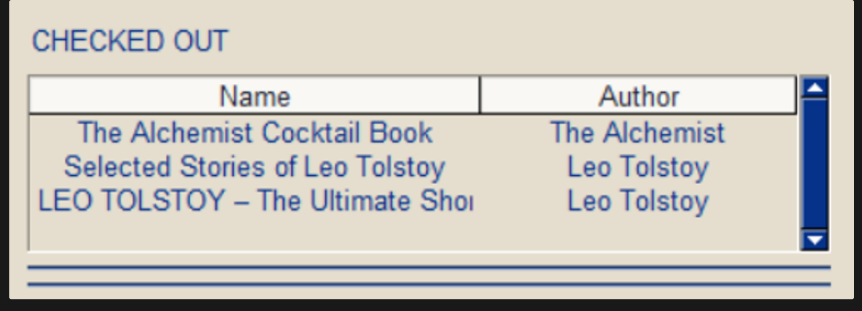


Home Page: After logging in, you'll land on the system's Home page. By default, the library is titled 'Harvard Central Library', but you can modify this `TITLE` in the `__init__.py` file located in the `classes` folder.

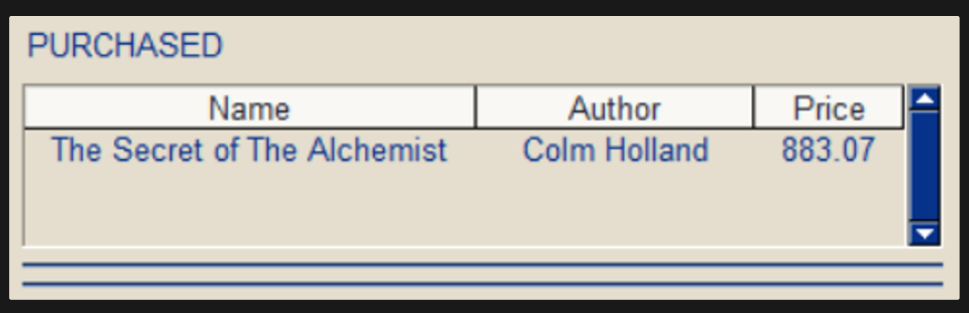
Checkouts: Displays the images of the checked-out books. Clicking on each book will trigger `Accounts.return()`. This action will check the book back in and deduct any late submission fines, if applicable.



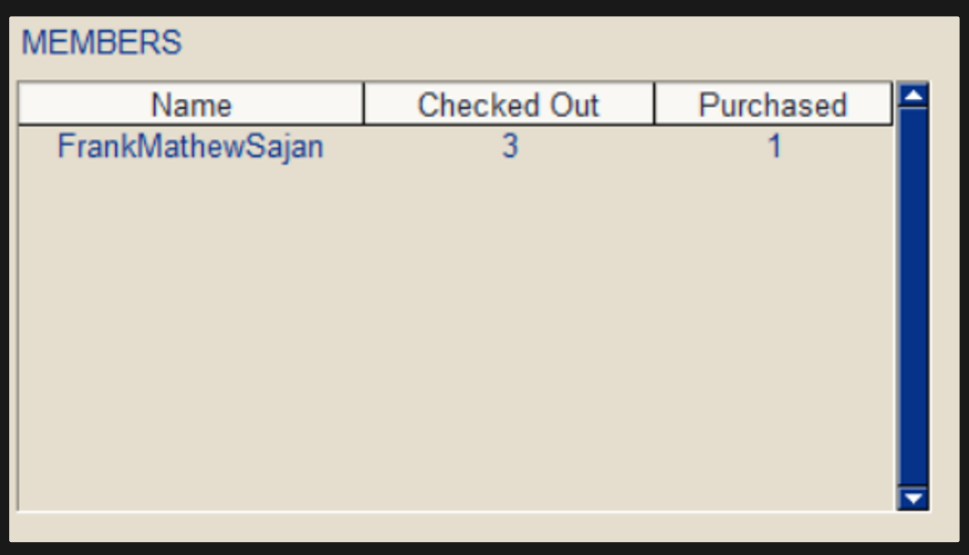
CHECKED OUT: Displays the name of the checked-out book and its author in a table.



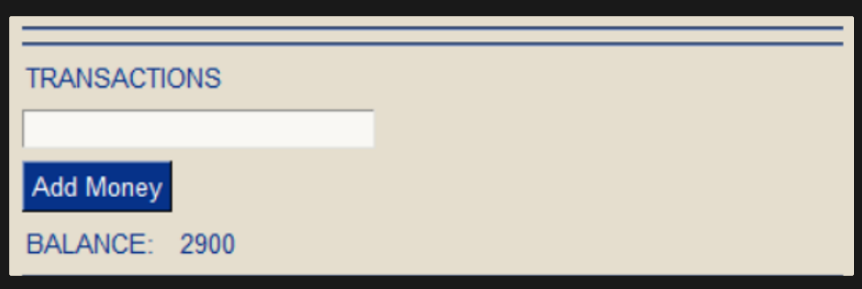
PURCHASED: Displays the name of the books purchased and their author on a table.



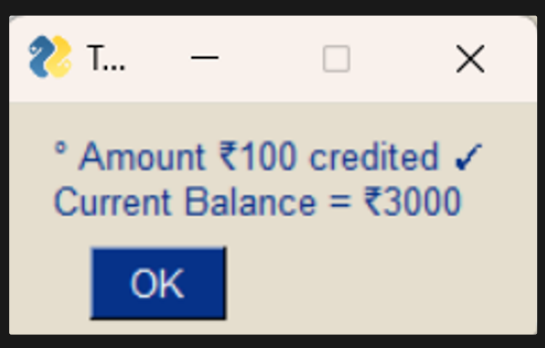
MEMBERS: Displays the name, number of books checked out, and number of books purchased by other members.



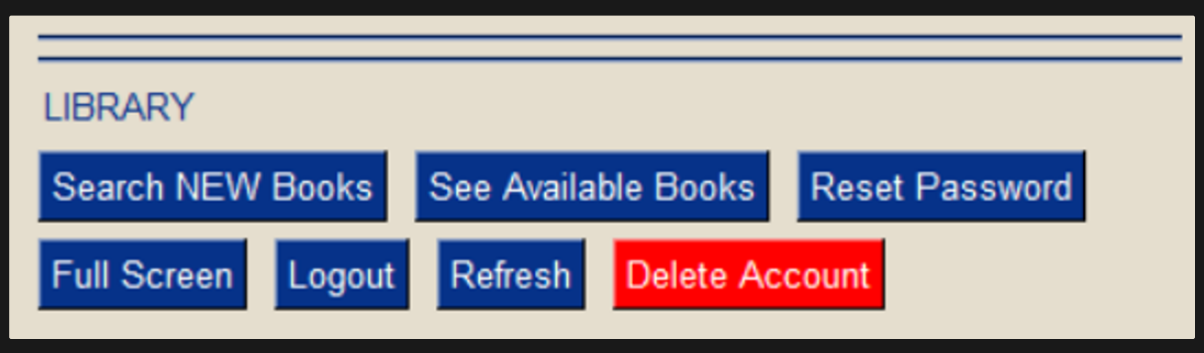
TRANSACTIONS:



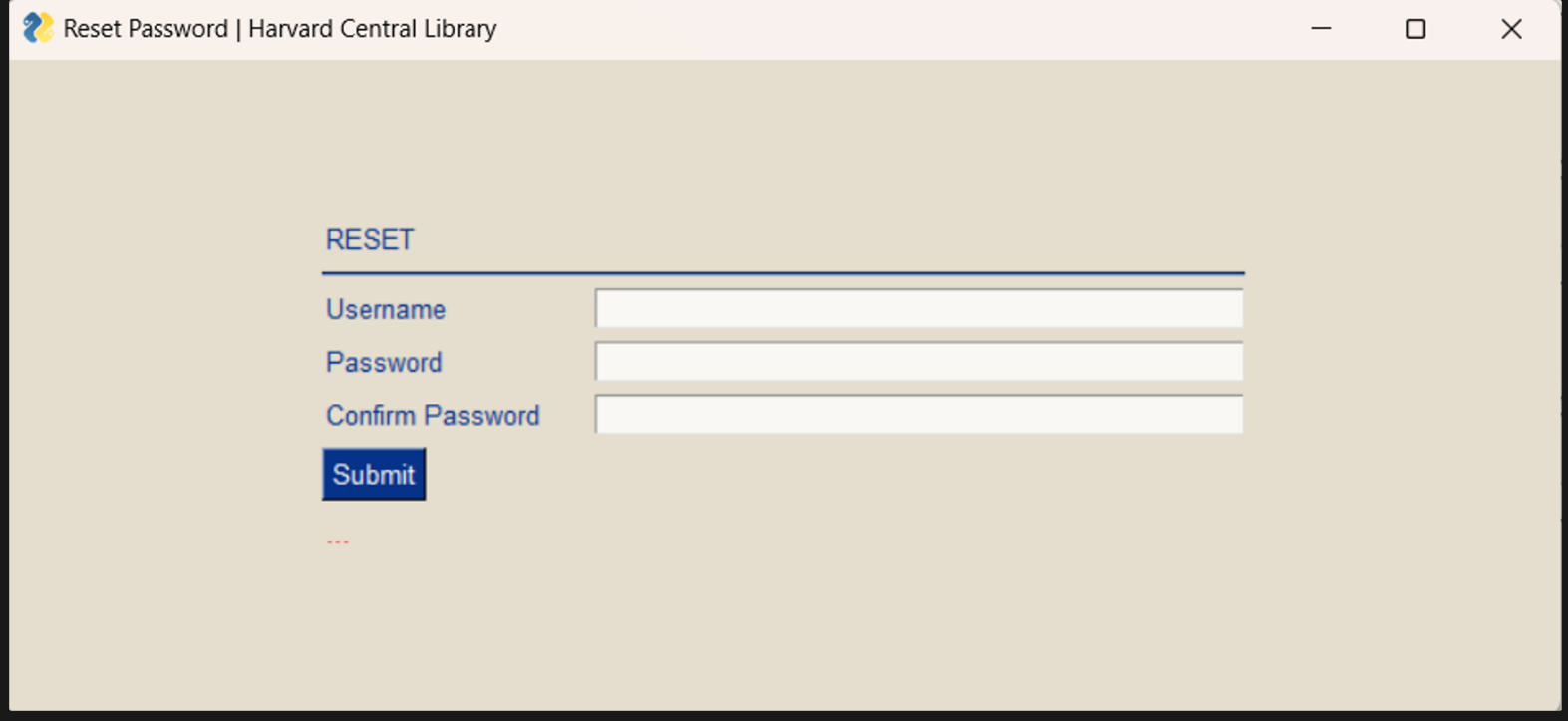
Enter the amount of money to add to the account in the input box. If the balance falls below zero due to any fines, all services, such as checkout, will be blocked until the balance is cleared.



LIBRARY SECTION



Reset Password: This opens the Reset Password Page.

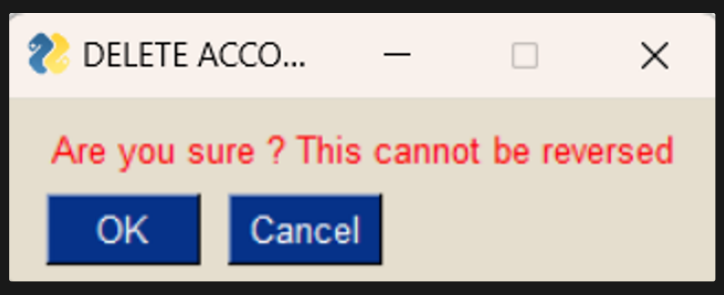


Full Screen: This maximizes the window.

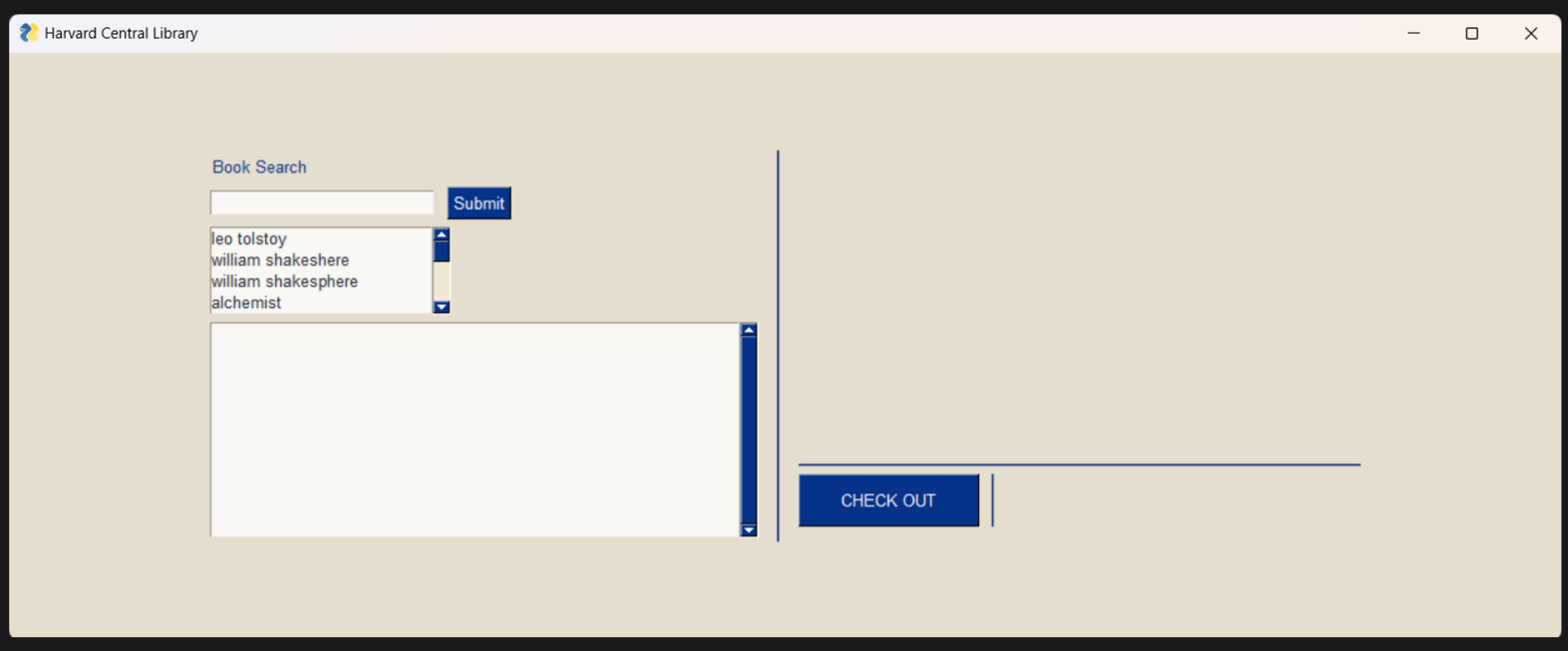
Logout: This logs out the current user.

Refresh: This refreshes the page with new information.

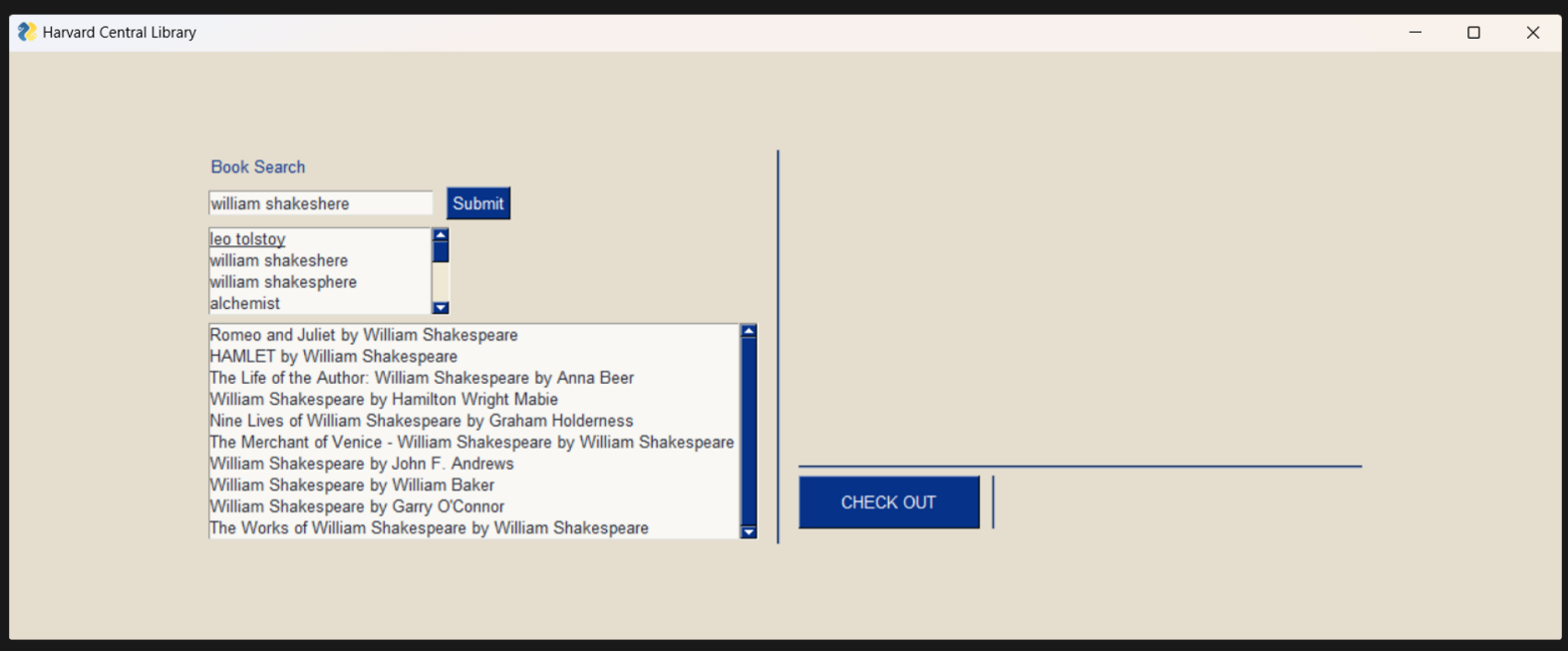
Delete Account: This deletes the account and its data.



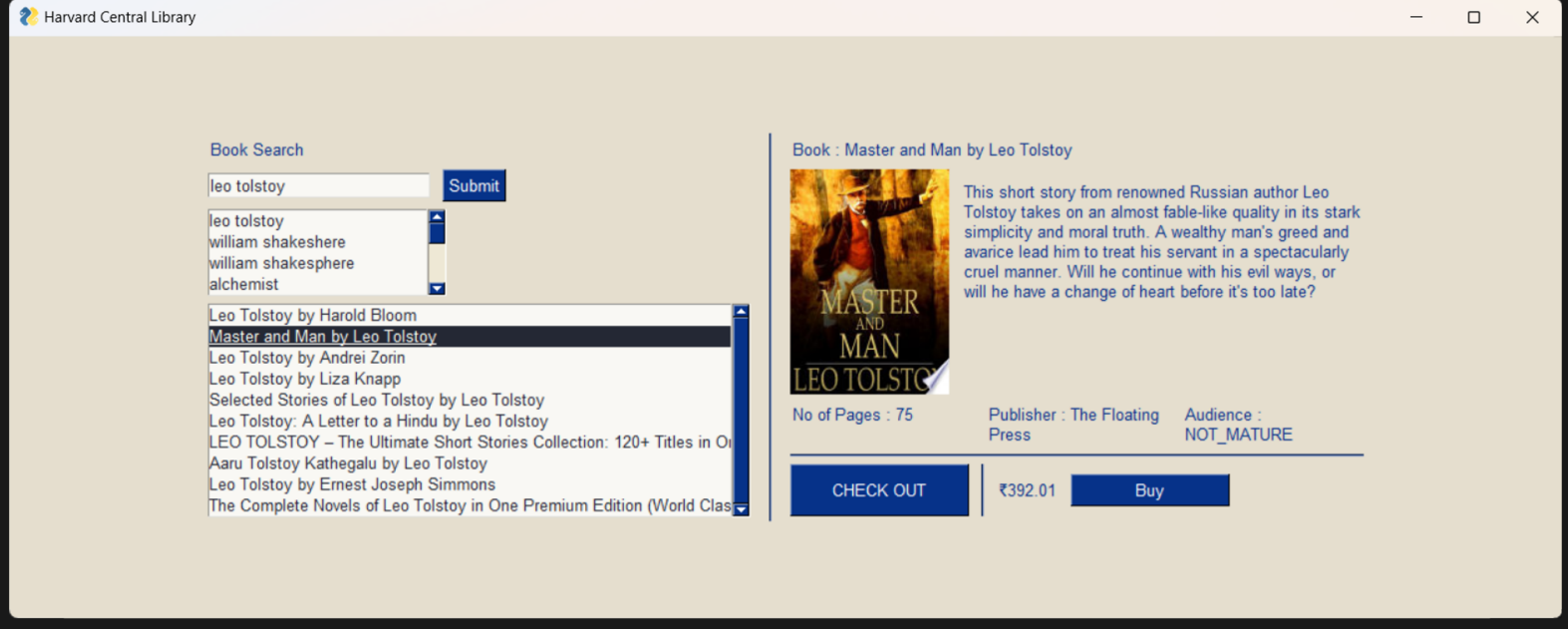
Search NEW Books & See Available Books:



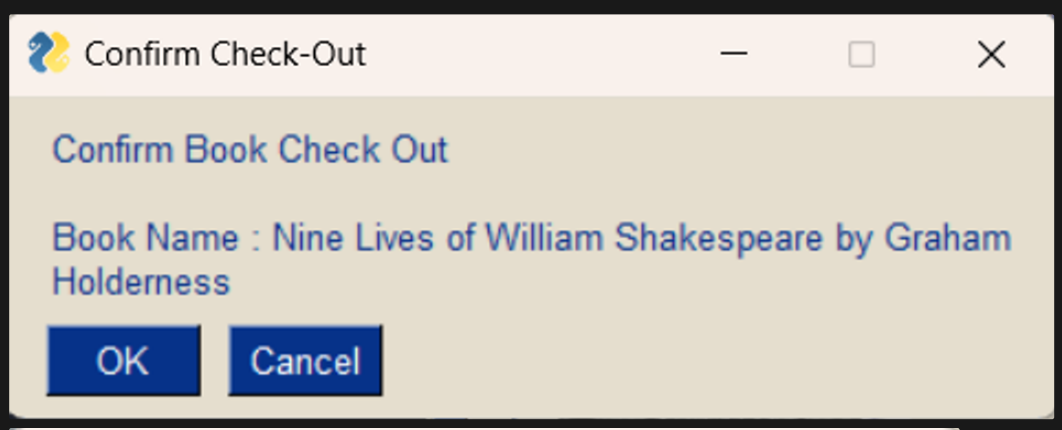
Both interfaces have the same GUI. The only difference is that `Search NEW Books` uses the `Google Books API` to search online. The small box below the search bar displays suggestions and recently searched books.

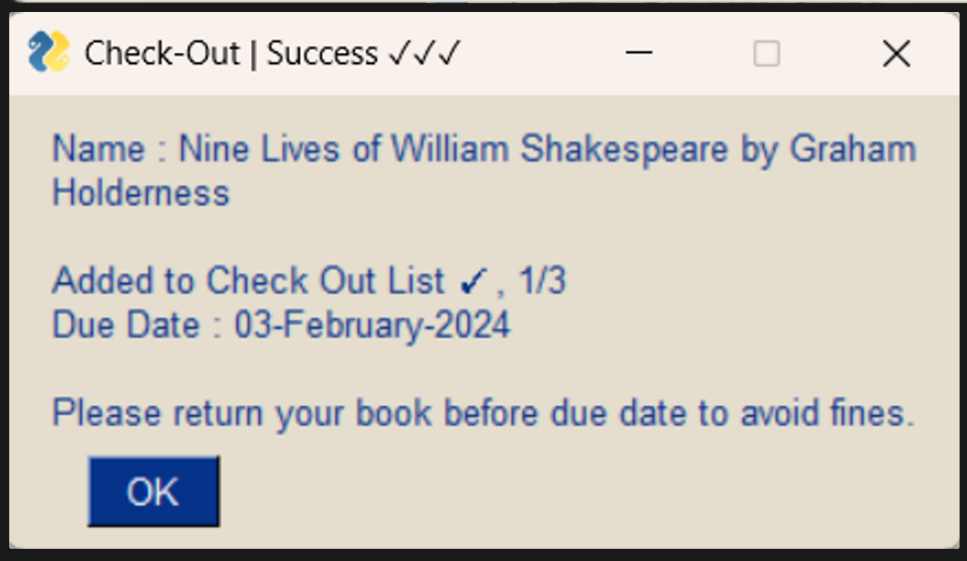


When the search button is pressed, a list of 10 books is displayed along with a preview. This preview includes a description, price, type, maturity rating, and the number of pages with the book preview, if the network is available.



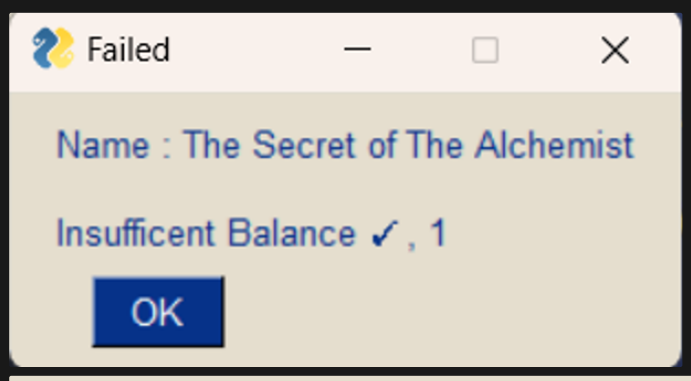
CHECKOUT: Check out the selected book.

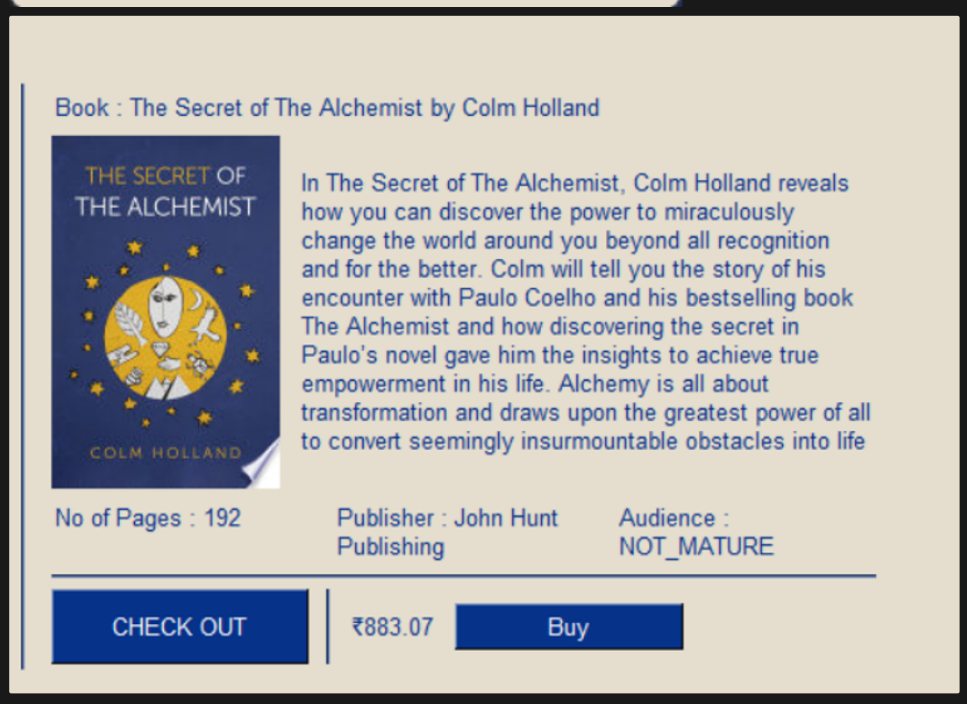


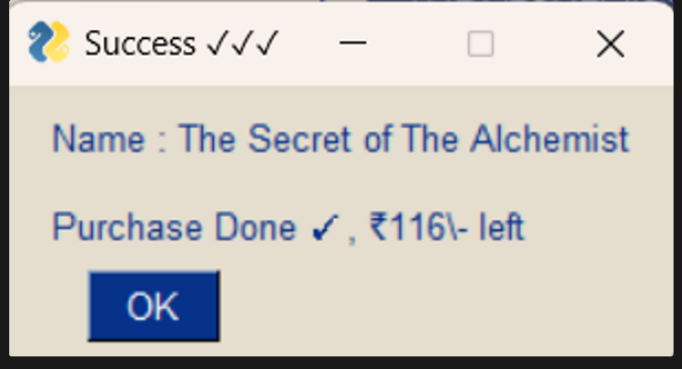


There is a checkout limit of 3 books. This can be changed in `CHECKOUT_LIMIT` `classes.__init__.py`

Buy: You can purchase a book if it's available for sale and you have sufficient balance.







Technical Side & Source Code

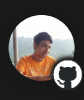
The project folder has the following structure.

```
| log.txt | project.py | README.md | requirements.txt | ├──assets | 1.png | 2.png | 3.png | image.png | ├──classes | Accounts.py | Database.py | Layouts.py | Library.py | __init__.py | ├──databases | sqlite.db
```

Constants and commonly use functions are saved in `classes.__init__.py`

```
AUTH = False TITLE = "Harvard Central Library" THEME = "TanBlue" ADMIN_PASS = b"$2b$12$RfGCPEzC5GwCCcnmpDIBhuZnZItstycBgpQWEqz/bqYcOJzsB9D9C" USERNAME_REGEX = r"^[a-zA-Z0-9]{3,20}$" USERNAME_CONDITIONS = """ Username Conditions 1. Be between 3 to 20 characters with no space in between. 2. Consist of only capital and small letters and numbers. 3. Not be only numbers """ PASSWORD_REGEX = r"^(?=.*\d)(?=.*[a-zA-Z])(?=.*[^\w\s]).{8,20}$" PASSWORD_CONDITIONS = """ Password Conditions 1. Be between 8 to 20 characters. 2. Be combination of letters, digits and special characters. 3. Atleast one digit, one alphabet and a special character (0-9A-Za-z ~!@#$%^&*). """ CHECKOUT_LIMIT = 3 DUE_DAYS = 10 FINE = 10 MAX_QUERIES = 10 #Functions are not displayed due to large number of code lines.
```

The entire source code is hosted on GitHub.



library-management-system
frankmathewsajan • Updated 17 days ago

frankmathewsajan/library-management-system (github.com)