How to run the file?

- The main code is stored in the “main.cpp” file and hence, **only** this file has to be executed.

- To run the file through terminal, type the following two commands in the terminal (after navigating to the folder where “main.cpp” and other .cpp files are stored) :

  g++ -o <name of the output file> main.cpp

  <name of the output file>

Catalog for each role

- **Student :**

  - **See all books**
  - **See books issued**
  - **Check whether you can issue another book or not**
  - **Issue a new book**
  - **Return a book**
  - **Search a book**
  - **Logout**

- **Professor :**

  - **See all books**
  - **See issued books**
  - **Issue a new book**
  - **Return a book**
  - **Search a book**
  - **Logout**

- **Librarian :**

  - **See all users**
  - **See all books**
  - **See which book is issued to which user**
  - **See which books have been issued**
  - **Add a new user**
  - **Delete a user**
  - **Add a new book**

- **Delete a book**
- **Search for a user**
- **Update details of a user**
- **Clear the book database**
- **Clear the user database**
- **Logout**

● Assumptions/ Program Flow

- A catalog has been prepared for each role which is viewed after login. Press the appropriate numeric key after login to perform the corresponding function. This catalog is displayed after each function is performed.
- As the code is run, the user is asked to login into his account. Initially, the database contains 5 users and 5 books. So, we must login into these 5 user accounts only initially
- Among the 5 users, there are 2 students, 2 professors and 1 librarian.
- Initial users look like following :

Name	Id	Password	Role	Username
Hitesh Anand	0	hitesh03	student	hitesh
Suresh Kumar	1	suresh1975	student	suresh
Indranil Saha	2	indranil74	professor	indranil
Subhajit Roy	3	iitkanpur202	professor	subhajit
Kritya Anand	4	kritya0110	librarian	kritya

- To add new users/books, view all the users/books, login as the librarian and choose appropriate options from the catalog.
- Student, Professor and Librarian classes inherit from the User class.

Class Description:

- **Book**

  - **print_details()** :- prints the details of the book
  - **check_issueable() :-** checks whether a book is available for issue or not
  - **get_title() :-** returns the title of the book
  - **get_author() :-** returns the author of the book
  - **get_publication() :-** returns the publication of the book

- **get_isbn() :-** returns the isbn of the book
- **change_issue_status() :-** changes the issue status of a book from “cannot be issued” to “can be issued”

- **BookDB**

  - **show_all_books() :** shows all books in the database
  - **add_book() :** add a book in the database
  - **get_count() :** number of books in the database currently
  - **delete_book(Book* b) :** delete a book from the database
  - **delete_book_with_title() :** delete a book with given title
  - **delete_book_by_isbn(string isbn) :** delete a book with given isbn
  - **clear_book_database() :** clear the whole book database
  - **search(Book* b) :** search for a book in the database
  - **search_by_isbn(string isbn) :** search for a book in the database with given isbn
  - **search_by_title(string title) :** search for a book in the database with given title
  - **search_by_author(string author) :** search for a book in the database with given author
  - **show_book_issued_by_user() :** show list of books issued along with their issuers
  - **show_all_issued_books() :** show the list of all issued books

- **User**

  - **get_name() :** returns name of the user
  - **get_id() :** returns id of the user
  - **get_password() :** returns password of the user
  - **get_username() :** returns username of the user
  - **set_name() :** sets a new name for the user
  - **set_role() :** sets a new role for the user
  - **set_pword() :** sets a new password for the user
  - **print_details() :** prints details for the user

- **UserDB**

  - **add_user() :** adds a user
  - **get_users_count() :** returns the number of users in the database
  - **show_all_users() :** show details of all the users in the database
  - **delete_user_by_uname() :** delete a user with given username
  - **search_by_uname() :** search user by username
  - **search_by_id() :** search user by id
  - **clear_user_database() :** clear user database

Limitations

- The database has been implemented using vectors and hence, the database gets refreshed each time the program restarts.