

**Department of Computer Engineering**

**Faculty of Technology, Dharmsinh Desai University**

**College Road, Nadiad - 387001**

**B. Tech. CE Semester –VI**

**Subject:**  SYSTEM DESIGN PRACTICE (CE-621)

**Project Title:** Library Management System

**Developed by:**

**Kashyap Tamakuwala**, CE-002, 17ITUBN022

**Riya Mistry**, CE-069, 17CEUOG003

**Gunjan Thumar**, CE-135, 17CEUOS081

**Guided by:**

**Prof. Hariom A. Pandya**

Assistant Professor, Department of Computer Engineering

Faculty of Technology, Dharmsinh Desai University, Nadiad

**Dharmsinh Desai University**

College Road, NADIAD - 387001. (Gujarat)



**CERTIFICATE**

*This is to certify that the project entitled “Library Management System” is a bonafide report of work carried out by*

*1) Mr. Kashyap Tamakuwala, Roll No: CE002, ID No: 17ITUBN022*

*2) Miss Riya Mistry, Roll No: CE069, ID No: 17CEUOG003*

*3) Mr. Gunjan Thumar, Roll No: CE135, ID No: 17CEUOS081*

*Of the Department of Computer Engineering, Semester VI, under the guidance and supervision for the subject System Design Practice. They were involved in the Project during the academic year 2019-2020.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Prof. Hariom A. Pandya Dr. C. K. Bhensdadia

Assistant Professor, Department Head of Department of

of Computer Engineering, Computer Engineering,

Dharmsinh Desai University, Dharmsinh Desai

Nadiad. University, Nadiad.

Date:

**CONTENTS**

1. Front Page …………………………………………………...01
2. Certificate …………………………………………………. .02
3. Abstract ……………………………………………………..04
4. Introduction ………………………………………………....05
5. Software Requirement Specification ………………………..06
6. Design ……………………………………………………….11
   1. ER Diagram …………………………………………..11
   2. Data Dictionary ……………………………………....12
7. Implementation Details ……………………………………..15
8. Screenshots …………………………………………………16
9. Conclusion ……………………………………………….....21
10. Limitation and Future Extension …………………………...22
11. Bibliography ………………………………………………..23

**Abstract**

Manual process of keeping student records, book records, account details, managing employee is very difficult.

There are various problems also faced by the student in a library such as finding any particular book, information whether the book is available or not, for what time this book will be available, searching for books using ISBN, etc.

To eliminate this manual system, library management system has been developed. Library Management System will handle all the current issues faced by the students and by its admin personnel.

**Introduction**

**Purpose**

Library Management System is a portal for efficient book management on behalf of the Ce department. It provides a graphical user interface that makes software interactive for both admin as well as the user.

**Technology Used:**

* Django
* Html
* Materialize Css
* Javascript

**Tools Used:**

* XAMPP Control Panel
* Git as VCS (Version Control System)
* Visual Studio Code as editor

**Platform:**

* Supported on all platform

**Software Requirement Specification for**

**Library Management System**

* **List of Users**:

1. Faculty
2. Student
3. Admin

* **Functional Requirements**

**1. General Requirements:**

R.1.1: Login:

**Description**: User Login with given id and password

**Input**: User Id and Password

**Output**: Show error if any or allows user to continue further

2. **Admin**

R.2.1: **Manage Books**

R.2.1.1: Add Book

**Description**: Admin can add Book

**Input**: Book details

**Output**: Message is shown.

R.2.1.2: Remove Books

**Description**: Admin can remove a book

**Input**: Book id

**Output**: Message is shown.

R.2.1.3: View issued Books

**Description**: Admin can see all the issued book

**Input**: Button click

**Output**: All issued books are shown.

R.2.2: **Manage Issue Request**

R.2.2.1: View Request

**Description**: Admin can see all the issue request for books

**Input**: Button click

**Output**: All Issue Request Are Shown

R.2.2.2: Accept Request

**Description**: Admin can accept an issue request

**Input**: Button click

**Output**: Book is issued successfully

R.2.2.3: Decline Request

**Description**: Admin can decline an issue request

**Input**: Button click

**Output**: Issue request is deleted

R.2.3: **Manage Members**

R.2.3.1: View Members

**Description**: Admin can see all the registered member

**Input**: Button click

**Output**: All Members Are Shown

R.2.3.2: Add a member

**Description**: Admin can add a member

**Input**: Details of the new member

**Output**: member is registered successfully

R.2.3.3: Remove a member

**Description**: Admin can remove a registered member

**Input**: Button click

**Output**: Member is unregistered

R.2.4: **Manage Overdue and Issue Period**

R.2.4.1: Change Issue Period

**Description**: Admin can change issue period of books

**Input**: New issue period

**Output**: Message is shown

R.2.4.2: Change Overdue Charge

**Description**: Admin can change the overdue charge

**Input**: New Value

**Output**: Message is Shown

R.2.4.3: Display charges for User

**Description**: Admin can see the list of user who have not returned book and can see late fee user need to pay

**Input**: Button click

**Output**: List is shown

R.2.4.3: Display charges for User

**Description**: Admin can see the list of user who have not returned book and can see late fee user need to pay

**Input**: Button click

**Output**: List is shown

R.2.4.3: Notify

**Description**: Admin can notify user of late submission and overdue fees.

**Input**: Button click

**Output**: Message is shown

3. **Faculty**

R.3.1: **Issuing**

R.3.1.1: Search Book

**Description**: user can search a book

**Input**: book name / Publisher Name/ Title

**Output**: List is shown

R.3.1.2: Issue Request

**Description**: user can request for issuing a book

**Input**: button click

**Output**: Message is Shown

R.3.1.3: Cancel issue Request

**Description**: User can cancel their issue request

**Input**: Button click

**Output**: Message is shown

R.3.1.4: View issued books

**Description**: User can see all books issued by him/her

**Input**: Button click

**Output**: List is shown

R.3.2: **Manage Details**

R.3.2.1: Change Password

**Description**: user change his/her Password

**Input**: new password

**Output**: message is shown

4. **Student**

R.4.1: **Issuing**

R.4.1.1: Search Book

**Description**: user can search a book

**Input**: book name / Publisher Name/ Title

**Output**: List is shown

R.4.1.2: Issue Request

**Description**: user can request for issuing a book

**Input**: button click

**Output**: Message is Shown

R.4.1.3: Cancel issue Request

**Description**: User can cancel their issue request

**Input**: Button click

**Output**: Message is shown

R.4.1.4: View issued books

**Description**: User can see all books issued by him/her

**Input**: Button click

**Output**: List is shown

R.4.1: **Manage Details**

R.4.2.1: Change Password

**Description**: user change his/her Password

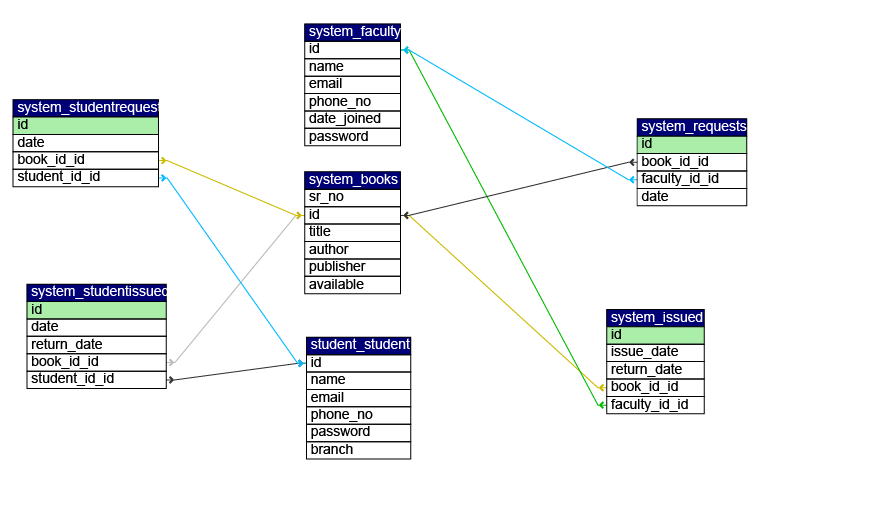
**Input**: new password

**Output**: message is shown

* **Non-functional Requirements:**
  + Secure access of confidential data (user’s details).
  + 24 X 7 availability
  + Better component design to get better performance at peak time
  + Flexible service based architecture will be highly desirable for future extension

**Design**

1. **E-R Diagram**



1. **Data Dictionary**

**BOOKS Table**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Field Name** | **Data Type** | **Width** | **Required** | **Unique** | **PK/FK** | | **Reference Table** | **Description** |
| 1 | Sr\_no | Int | 4 | yes | Yes | | PK | - | Numbering for Books |
| 2 | Id | varchar | 100 | yes | Yes | | - | - | Unique ID for Books |
| 3 | Title | Varchar | 100 | Yes | No | | - | - | Title of Book |
| 4 | Author | Varchar | 100 | Yes | No | | - | - | Author of the Book |
| 5 | Publisher | Varchar | 100 | Yes | No | | - | - | Publisher of Book |
| 6 | Available | Int | 10 | Yes | NO | | - | - | Boolean flag for denoting issued or not |

**Faculty Table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Field Name** | **Data Type** | **Width** | **Required** | **Unique** | **PK/FK** | **Reference Table** | **Description** |
| 1 | Id | Varchar | 100 | yes | yes | PK | - | Faculty id |
| 2 | Name | Varchar | 100 | yes | No | - | - | Faculty Name |
| 3 | Email | Varchar | 100 | Yes | No | - | - | Email address |
| 4 | Phone Number | Int | 10 | Yes | No | - | - | Phone Number |
| 5 | Date Joined | Date | 100 | Yes | No | - | - | Date of Joining |
| 6 | Password | Varchar | 10 | Yes | No | - | - | - |

**Student Table**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Field Name** | **Data Type** | **Width** | **Required** | | **Unique** | | **PK/FK** | | **Reference Table** | **Description** |
| 1 | Id | Varchar | 100 | | yes | | yes | | PK | - | Student id |
| 2 | Name | Varchar | 100 | | yes | | No | | - | - | Student Name |
| 3 | Email | Varchar | 100 | | Yes | | No | | - | - | Email address |
| 4 | Phone Number | Int | 10 | | Yes | | No | | - | - | Phone Number |
| 5 | Branch | Varchar | 100 | | Yes | | No | | - | - | Branch of Student |
| 6 | Password | Varchar | 10 | | Yes | | No | | - | - | - |

**Faculty Issue Table**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Field Name** | **Data Type** | **Width** | **Required** | | **Unique** | | **PK/FK** | | **Reference Table** | **Description** |
| 1 | Id | Int | 4 | | yes | | Yes | | PK | - | id |
| 2 | Issue-Date | Date | 4 | | Yes | | yes | | - | - | - |
| 3 | Return-Date | Date | 4 | | Yes | | Yes | | - | - | - |
| 4 | Book\_id\_id | Varchar | 5 | | Yes | | No | | FK | Books Table | Id of the Book |
| 5 | Faculty\_id\_id | Varchar | 10 | | Yes | | No | | FK | Faculty Table | Id of the Faculty |

**Faculty Request Table**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Field Name** | **Data Type** | **Width** | **Required** | | **Unique** | | **PK/FK** | | **Reference Table** | **Description** |
| 1 | Id | Int | 4 | | yes | | Yes | | PK | - | id |
| 2 | Date | Date | 4 | | Yes | | yes | | - | - | - |
| 3 | Book\_id\_id | Varchar | 5 | | Yes | | No | | FK | Books Table | Id of the Book |
| 4 | Faculty\_id\_id | Varchar | 10 | | Yes | | No | | FK | Faculty Table | Id of the Faculty |

**Student Issue Table**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Field Name** | **Data Type** | **Width** | **Required** | | **Unique** | | **PK/FK** | | **Reference Table** | **Description** |
| 1 | Id | Int | 4 | | yes | | Yes | | PK | - | id |
| 2 | Issue-Date | Date | 4 | | Yes | | yes | | - | - | - |
| 3 | Return-Date | Date | 4 | | Yes | | Yes | | - | - | - |
| 4 | Book\_id\_id | Varchar | 5 | | Yes | | No | | FK | Books Table | Id of the Book |
| 5 | Student\_id\_id | Varchar | 10 | | Yes | | No | | FK | Student Table | Id of the Student |

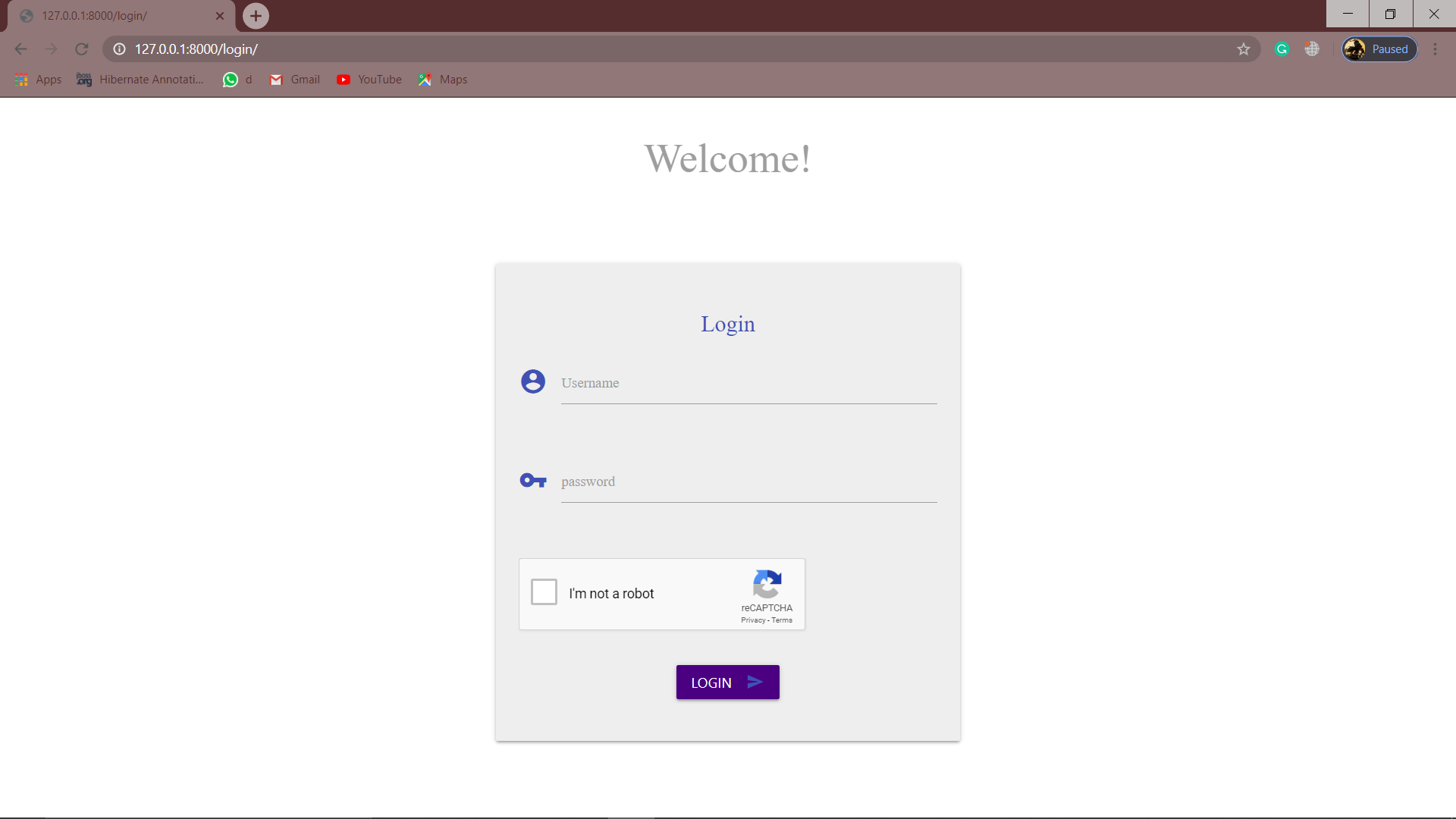
**Student Request Table**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Field Name** | **Data Type** | **Width** | **Required** | | **Unique** | | **PK/FK** | | **Reference Table** | **Description** |
| 1 | Id | Int | 4 | | yes | | Yes | | PK | - | id |
| 2 | Date | Date | 4 | | Yes | | yes | | - | - | - |
| 3 | Book\_id\_id | Varchar | 5 | | Yes | | No | | FK | Books Table | Id of the Book |
| 4 | Student\_id\_id | Varchar | 10 | | Yes | | No | | FK | Student Table | Id of the Student |

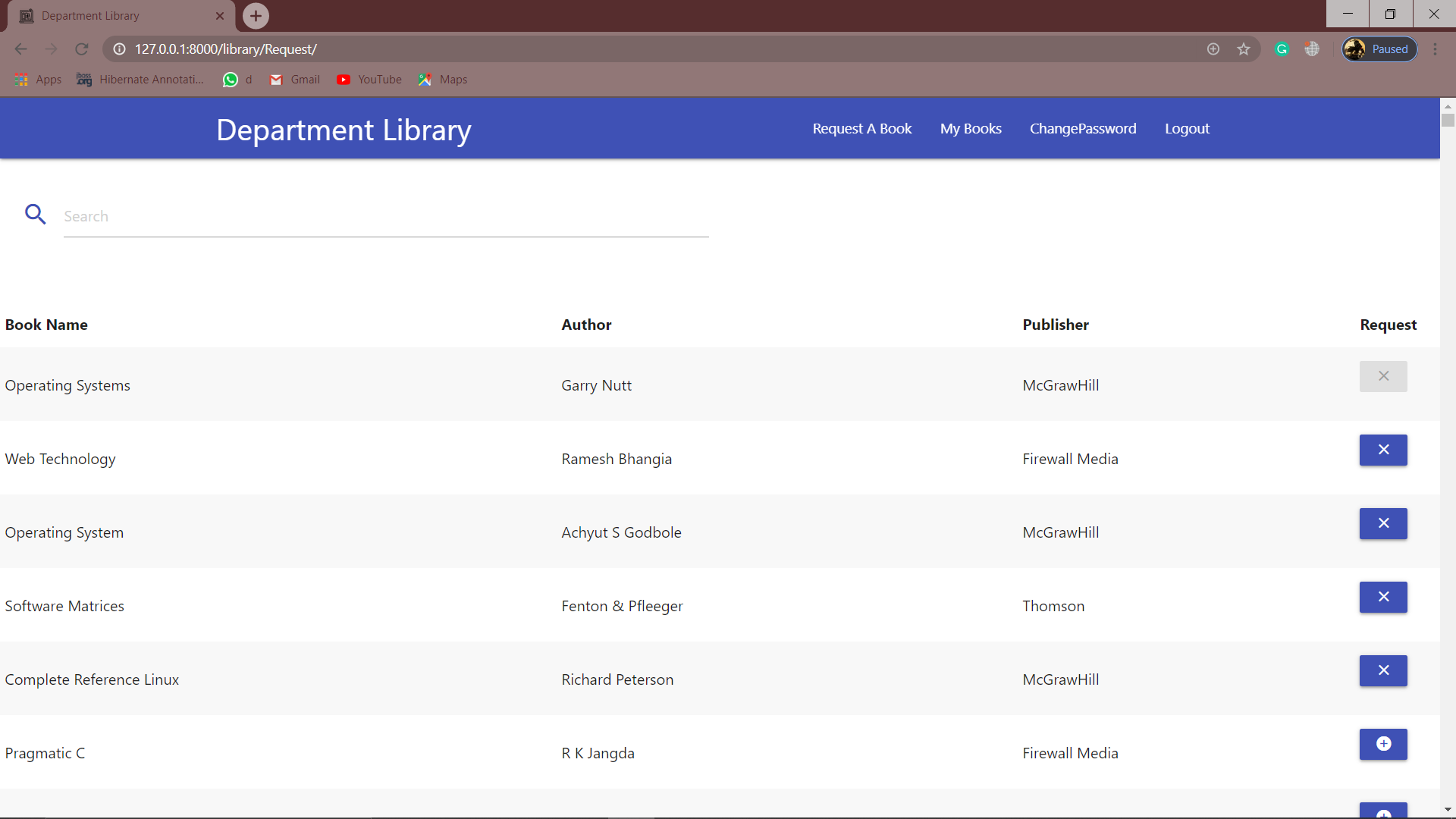
**Implementation Details**

* Implementation of all apps follows:
  + MVC (Model View Controller)
* Almost all of the codebase follows the **SOLID Principle.**
* MySQL database used for database query that is Relational DBMS(RDBMS).
* To Design All of the UI Html, Materialize CSS and JavaScript is used throughout the project
* Users of this system are Admin, Faculty, and Student.
  + To manage users, while registering it the system assigns roles to users as per their input.
  + Next time at login time, Django Session comes to picture and it will show content according to the user.
  + Overdue Fines are visible to user accordingly.
  + Only Admin is able to add a book, faculty or student.

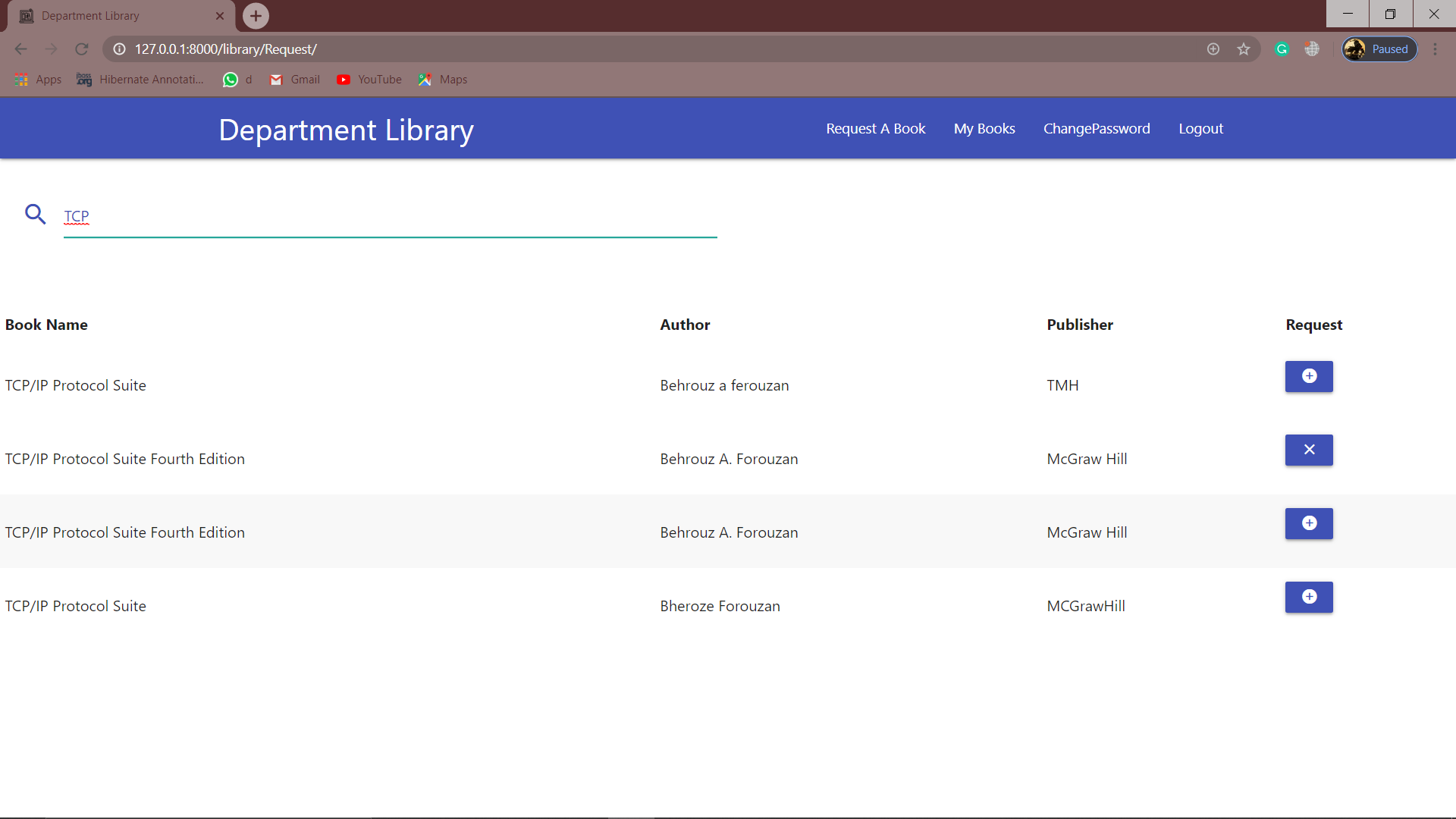
**Screenshot**

****

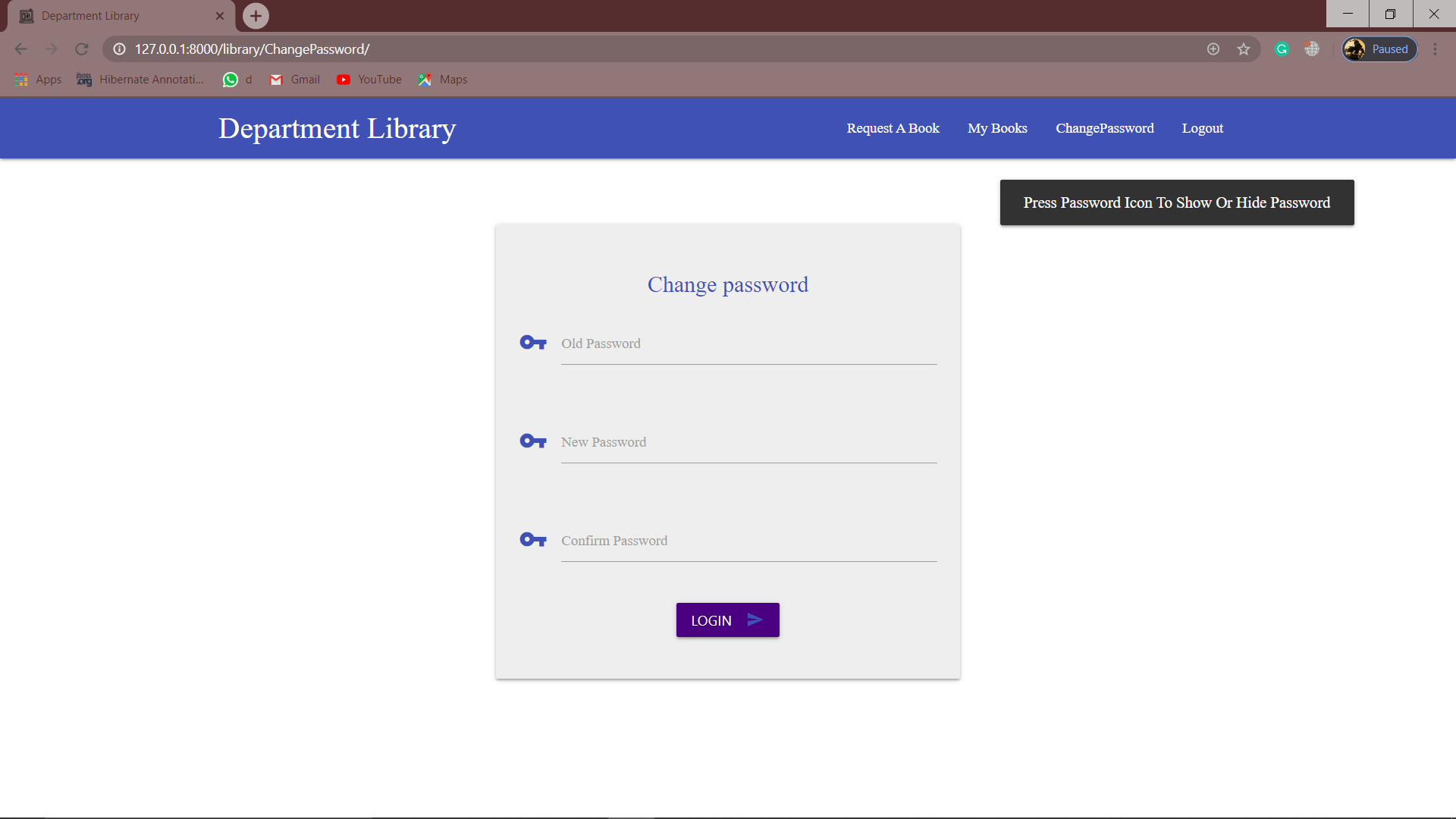
1. logging page



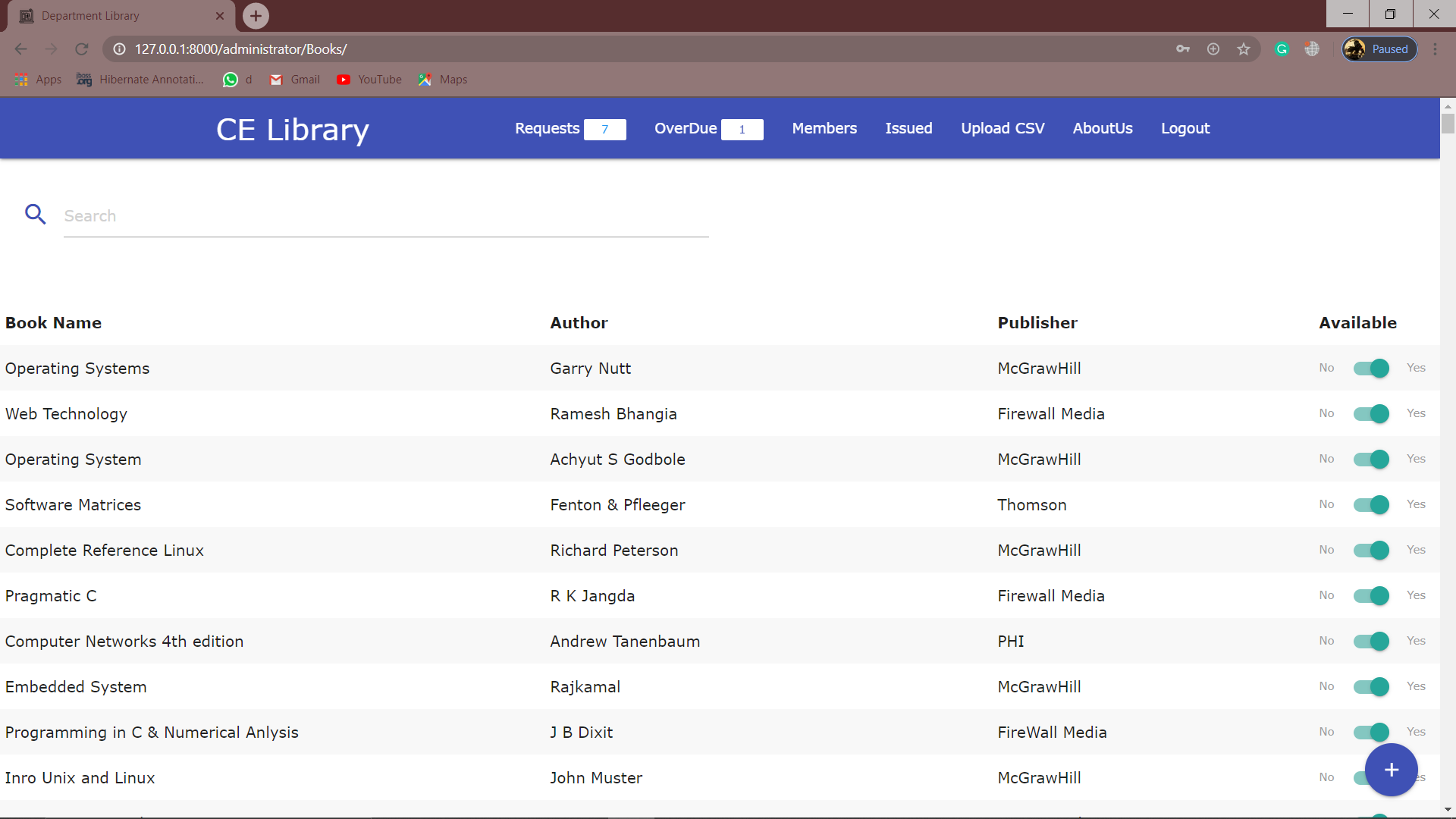
II) Faculty/Student Homepage



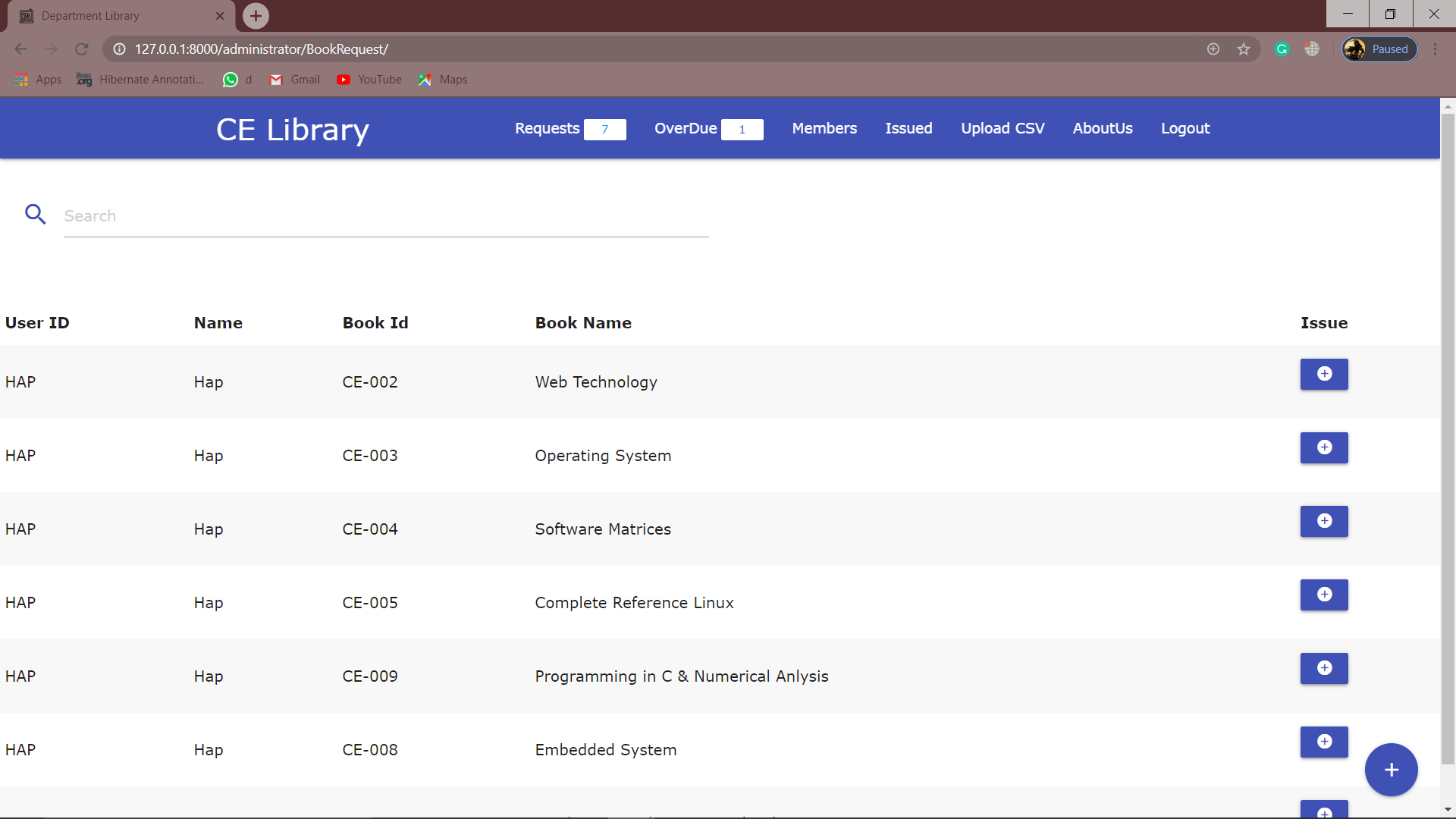
III) Search Book



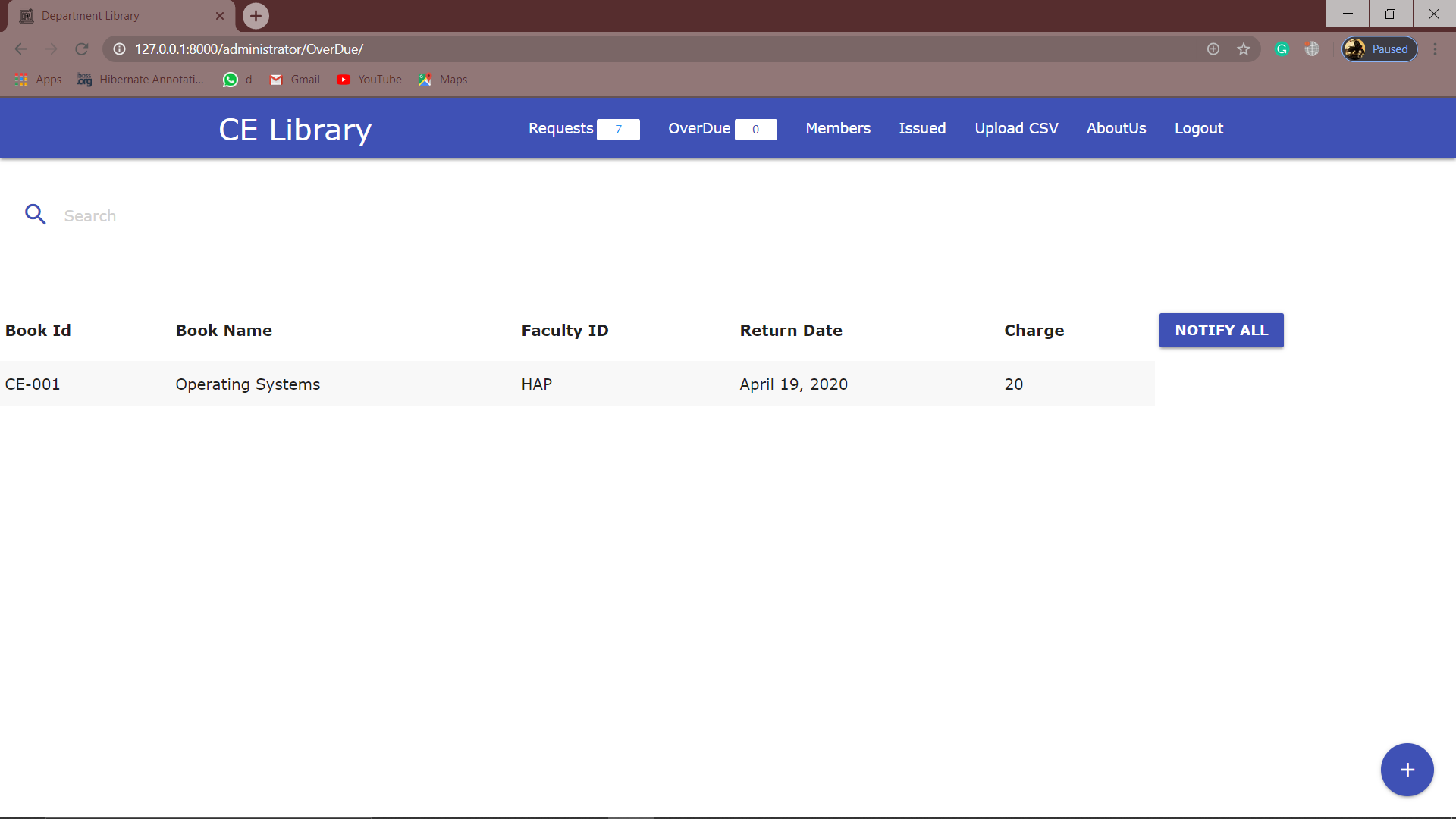
IV) Change Password



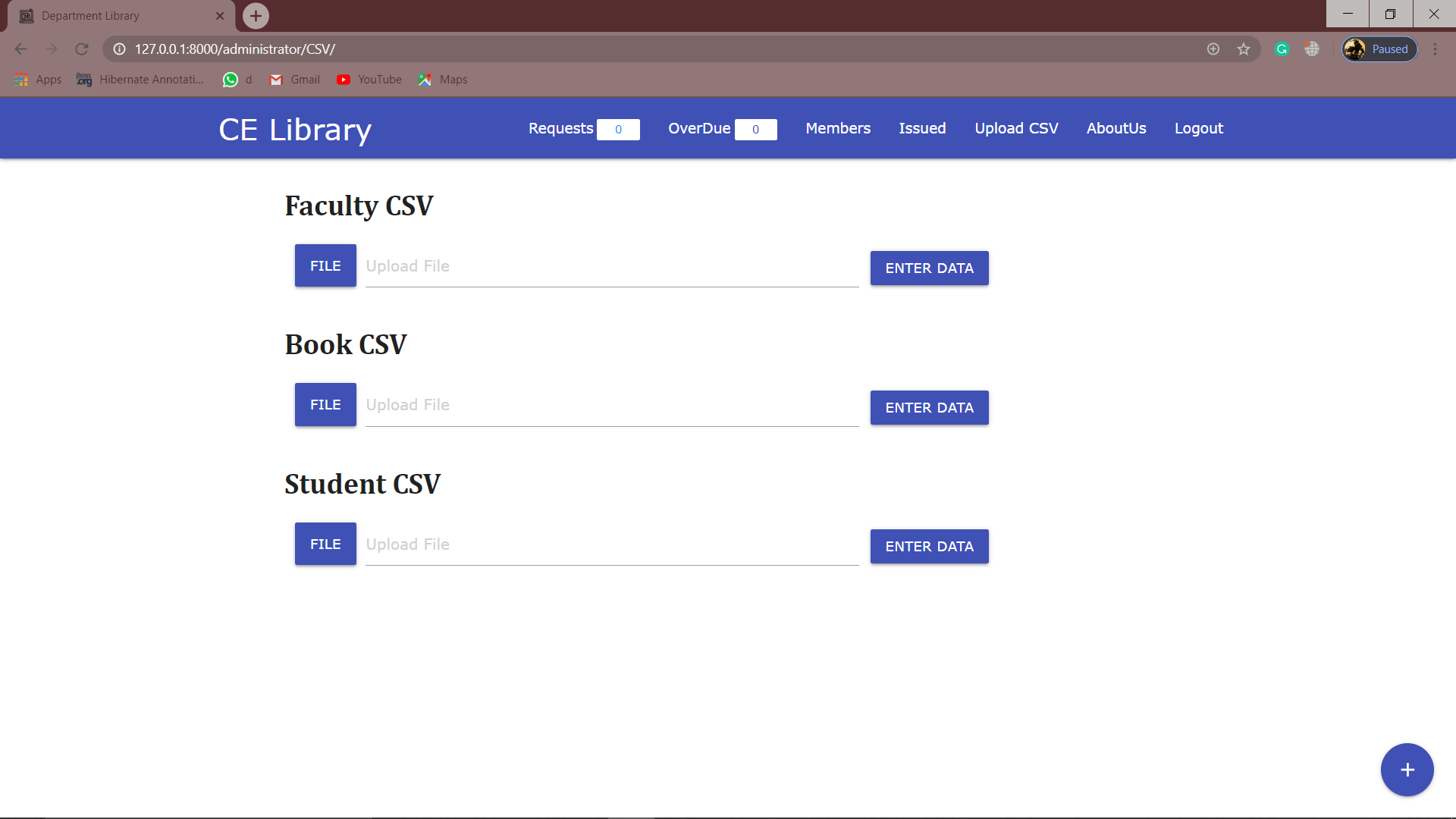
V) Admin Homepage



VI) Issue Request



VII) Overdue Page



VIII) Upload CSV (For adding data from csv file)

**Conclusion**

* We have implemented all the necessary functionalities for library management.
* We have also taken care of Security issues like Csrf (Cross Site Reference Forgery).
* Users can search books effectively and can issue a book.
* The application is mostly dynamic so that all data is rightful for users.
* Application aims to give effective user experience to end-users.

**Limitation and Future extension**

**Future Extension**

* Improving UI.
* Extension in web app.
* Implementation of Qr Scanner or Bar Code Scanner for Books

**Bibliography**

* To Implement Project Successfully we referred many websites, blogs, forums.
* Reference Links:
  + <https://materializecss.com/>
  + <https://stackoverflow.com/>
  + <https://www.tutorialspoint.com/>
  + <https://docs.djangoproject.com/en/3.0/howto/deployment/>